Expanding the Compute-and-Forward Framework: Unequal Powers, Signal Levels, and Multiple Linear Combinations

Bobak Nazer, *Member, IEEE,* Viveck Cadambe, *Member, IEEE,*
Vasilis Ntranos, *Member, IEEE,* and Giuseppe Caire, *Fellow, IEEE*

***Abstract*—The compute-and-forward framework permits each receiver in a Gaussian network to directly decode a linear combination of the transmitted messages. The resulting linear combinations can then be employed as an end-to-end communication strategy for relaying, interference alignment, and other applications. Recent efforts have demonstrated the advantages of employing unequal powers at the transmitters and decoding more than one linear combination at each receiver. However, neither of these techniques fit naturally within the original formulation of compute-and-forward. This paper proposes an expanded compute-and-forward framework that incorporates both of these possibilities and permits an intuitive interpretation in terms of signal levels. Within this framework, recent achievability and optimality results are unified and generalized.**

***Index Terms*—interference, nested lattice codes, compute-and-forward, successive decoding, unequal powers**

I. Introduction

Consider a Gaussian wireless network consisting of multiple transmitters and receivers. In this context, the compute-and-forward framework of [1] enables the receivers to decode linear combinations of the messages, often at much higher rates than what would be possible for decoding the individual messages. This strategy can be used as a building block for relaying strategies [2], MIMO integer-forcing transceiver architectures [3]–[8], or interference alignment schemes [9]–[11].

The coding scheme underlying this compute-and-forward framework maps the messages, which are viewed as elements of a vector space over a prime-sized finite field, to nested lattice codewords. The receivers are then able to decode integer-linear combinations of the codewords with coefficients chosen to approximate the real-valued channel coefficients (with better approximations yielding higher rates). Finally, these integer-linear combinations of lattice codewords are mapped back to the vector space over the finite field to yield linear combinations of the original messages. In other words, compute-and-forward creates a direct connection between network coding over a finite field and signaling over a Gaussian channel.

We now recall two simple properties of Gaussian networks with multiple transmitters and receivers. First, the amount of available power may vary across transmitters. Second, the noise variance may vary across receivers. Thus, we would like our coding scheme to be versatile enough to allocate power unequally across transmitters as well as space codewords far enough apart to tolerate the noise at the targeted receivers. For classical random coding strategies that aim to deliver subsets of the messages to the receivers, these two forms of versatility can be viewed as simply the flexibility to adjust the targeted signal-to-noise ratio (SNR) for each codeword. However, in the compute-and-forward setting, the receivers want linear combinations of the messages, and the effect of noise variance and power on the nested lattice codebooks is more nuanced than in the classical random coding setting. In particular, the codeword spacing for a given message is determined by the maximum noise variance across all receivers whose desired linear combinations involve the message. This codeword spacing corresponds to the density of the fine lattice from which the codewords are drawn. Additionally, the power level for a given message is determined by the power constraint of the associated transmitter. This power level corresponds to the second moment of the coarse lattice used for shaping.

The goal of this paper is to expand the compute-and-forward to include these two forms of versatility while retaining the connection between the finite field messages and the lattice codewords. Prior work has focused on either varying the noise tolerances (i.e., the codeword spacings) or the power levels but not both. For instance, the framework in [1] permits the codewords to tune their noise tolerances but requires that the codewords have the same power level. In [12], Nam *et al.* proposed a nested lattice technique that permits unequal power levels for multiple transmitters that communicate the sum of their codewords to a single receiver over a symmetric channel (i.e., all channel gains are equal to one). However, this technique does not establish a connection between messages drawn from a finite field and lattice codewords. Part of the motivation for this paper is to unify the techniques from [1] and [12] into a single framework.

The primary contribution of this paper is an *expanded compute-and-forward framework* that permits both unequal

B. Nazer was supported by NSF grant CCF-1253918. This work was presented at the 51st Allerton Conference on Communications, Control, and Computing in 2013 and the 10th Information Theory and Applications Workshop in 2015.

B. Nazer is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA. Email: bobak@bu.edu.

V. Cadambe is with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA. Email: viveck@engr.psu.edu.

V. Ntranos is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA. Email: ntranos@usc.edu.

G. Caire is with the Department of Telecommunication Systems, Technical Universität Berlin, Berlin, Germany and the Department of Electrical Engineering, University of Southern California, Los Angeles, CA. Email: caire@tu-berlin.de.

powers and noise tolerances across transmitters while providing a mapping between finite field messages and lattice codewords. Interestingly, this framework allows us to interpret both the power constraint and noise tolerance associated to each message in terms of "signal levels," in a manner reminiscent of the deterministic model of Avestimehr *et al.* [13]. Specifically, each transmitter's message is a vector from $\mathbb{F}_p^k$ where p is prime. The power level of the transmitter determines a "ceiling" above which the message vector must be zero. Similarly, the noise tolerance of the transmitter determines a "floor" below which the message vector must be zero. The information symbols of the transmitter are placed between these constraints.

Recent work has studied the problem of recovering multiple linear combinations at a single receiver. In particular, Feng *et al.* [14] linked this problem to the shortest independent vector problem [15] and a sequence of papers has demonstrated its value for integer-forcing MIMO decoding [3], [6]–[8] as well as for integer-forcing interference alignment [9]–[11]. However, the original compute-and-forward framework does not capture some of the subtleties that arise when decoding multiple linear combinations. For instance, as shown by Ordentlich *et al.* [9], after one or more linear combinations have been decoded, they can be used as side information to eliminate some of the codewords from subsequent linear combinations. This *algebraic successive cancellation* technique eliminates some of the rate constraints placed on codewords, i.e., it enlarges the rate region. Also, as shown in [9], this technique can be used to approach the multiple-access sum capacity within a constant gap. Additionally, recent work by the first author [16] as well as Ordentlich *et al.* [7] revealed that decoded linear combinations can be used to infer the corresponding integer-linear combination of channel inputs, which can in turn be used to reduce the effective noise encountered in subsequent decoding steps. As argued in [7], this *successive computation* technique can reach the exact multiple-access sum capacity.

Our expanded compute-and-forward framework is designed with multiple linear combinations in mind. Specifically, we use a *computation rate region* to capture the dependencies between rate constraints. Our achievability results broaden the algebraic successive cancellation and successive computation techniques to permit unequal powers as well as scenarios where the number of messages exceeds the number of desired linear combinations. We capture the prior results of [3], [7], [9], [16] as special cases and shed additional light on the structure of optimal integer matrices for successive decoding.

Beyond unifying existing results, this expanded framework is meant to serve as a foundation for ongoing and future applications of compute-and-forward and integer-forcing techniques. For example, the initial motivation behind developing this framework was the authors' exploration of integer-forcing for interference alignment [10]. Subsequently, He *et al.* [17] have used this framework to propose a notion of uplink-downlink duality for integer-forcing and Lim *et al.* [18] used it to propose a discrete memoryless version of compute-and-forward.

A. Related Work

The main concept underlying compute-and-forward is that the superposition property of the wireless medium can be exploited for network coding [19]–[21]. This phenomenon was independently and concurrently discovered by [22]–[24], with the latter coining the phrase *physical-layer network coding.* Subsequent efforts [12], [25], [26] developed lattice coding strategies for communicating the sum of messages to a single receiver. This lead to the compute-and-forward framework [1] for multiple receivers that recover linear combinations of the messages (albeit with equal power constraints, unlike the single receiver framework of [12]).

As shown by Feng *et al.* [14], any compute-and-forward scheme based on nested lattice codes can be connected to network coding over a finite commutative ring. From this algebraic perspective, the compute-and-forward framework of [1] can be viewed as a special case that connects nested lattice codes generated via Construction A to network coding over a prime-sized finite field. Another important special case is the recent work of Tunali *et al.* [27] that develops a compute-and-forward scheme based on nested lattices over Eisenstein integers. For complex-valued channels, this scheme can offer higher computation rates on average (e.g., for Rayleigh fading) since the Eisenstein integers are a better covering of the complex plane than the Gaussian integers employed by [1]. Several recent papers have also used the algebraic perspective of [14] to propose practical codes and constellations for compute-and-forward [28]–[31].

The line of work on compute-and-forward is part of a broader program aimed at uncovering the role of algebraic structure in network information theory, inspired by the paper of Körner and Marton [32]. For instance, there are advantages for using algebraic structure in coding schemes for dirty multiple-access [33]–[35], distributed source coding [36]–[40], relaying [2], [5], [41]–[44], interference alignment [9], [10], [45]–[49], and physical-layer secrecy [50]–[53]. Many of these works benefit from the development of lattice codes that are good for source and channel coding as well as binning [54]–[59]. We refer readers to the textbook of Zamir [60] for a full history of these developments, an in-depth look at lattice constructions, achievable rates, and applications as well as a chapter [61] on the use of lattice codes in network information theory.

In recent, independent work, Zhu and Gastpar [62] proposed a compute-and-forward scheme for unequal powers based on the scheme of [12]. They also showed how to use this scheme to reach the two-user multiple-access sum capacity (if the channel strength lies above a small constant). However, their scheme does not retain the connection to a finite field.

The original motivation for the compute-and-forward strategy was the possibility of relaying in a multi-hop network while avoiding the harmful effects of interference between users (by decoding linear combinations) as well as noise accumulation (by decoding at every relay). Several works have investigated and improved upon the performance of the original compute-and-forward framework in the context of multi-hop relaying [41], [43], [48], [63]–[65]. Our expanded framework can improve performance further by permitting relays to employ unequal powers and decode multiple linear combinations when appropriate.[1]

B. Paper Organization

We have strived to present our results, some of which are rather technical, in an accessible fashion. To this end, we begin in Section II with an informal overview of our framework to build intuition, before giving a formal problem statement. We then state our main results in Section III without using any lattice definitions or properties. Afterwards, we introduce our nested lattice code construction in Section IV and proceed to prove our main achievability theorems in Sections V and VI.

II. Problem Statement

In this section, we provide a problem statement for the expanded compute-and-forward framework. As mentioned earlier, our message structure can be interpreted in terms of signal levels that resemble the deterministic model of Avestimehr *et al.* [13].[2] Unlike the original compute-and-forward problem of [1], we will not aim to directly decode linear combinations of the messages. Instead, we associate each message realization with a coset and aim to decode linear combinations of vectors that belong to the same cosets as the transmitted messages. We describe a simple method of understanding this class of linear combinations through the use of "don't care" entries. As we will argue, if the coefficients of the linear combinations would suffice to recover a given subset of the messages in the original compute-and-forward framework [1], they also suffice to recover these messages in the expanded compute-and-forward framework. For example, if the receiver obtains a full-rank set of linear combinations, all of the transmitted messages can be recovered successfully.

For the sake of conciseness, we will focus on real-valued channel models with additive white Gaussian noise (AWGN). Our coding theorems can be applied to complex-valued channel models either via a real-valued decomposition of the channel [1], [3] or by building nested lattice codebooks directly over the complex field using either Gaussian or Eisenstein[3] integers [14], [68]. While our framework is intended for AWGN networks with any number of sources, relays, and destinations, we find it clearer to first state our main results from the perspective of a single receiver that wishes to decode one or more linear combinations. In Section III-D, we will show how to apply our coding theorems to scenarios with multiple receivers. Below, we state our notational conventions, essential definitions for our channel model, a high-level overview of the compute-and-forward problem, and a formal problem statement.

A. Notation

We will employ the following notation. Lowercase, bold font (e.g., $\mathbf{x}$) will be used to denote column vectors and uppercase, bold font (e.g., $\mathbf{H}$) will be used to denote matrices. For any matrix $\mathbf{H}$, we denote the transpose by $\mathbf{H}^{\mathsf{T}}$, the span of its rows by $\mathrm{rowspan}(\mathbf{H})$, and its rank by $\mathrm{rank}(\mathbf{H})$. The notation $\|\mathbf{x}\|$ denotes the Euclidean norm of the vector $\mathbf{x}$ while $\lambda_{\text{min}}(\mathbf{H})$ and $\lambda_{\text{max}}(\mathbf{H})$ denote the minimum and maximum singular values of $\mathbf{H}$, respectively. We will denote the all-zeros column vector of length k by $\mathbf{0}_k$, the $k \times n$ all-zeros matrix by $\mathbf{O}_{k\times n}$, the all-ones column vector of length k by $\mathbf{1}_k$, and the $k \times k$ identity matrix by $\mathbf{I}_k$. We will sometimes drop the subscript when the size can be inferred from the context. The $\log$ operation will always be taken with respect to base 2 and we define $\log^+(x) = \max(0, \log(x))$.

Our framework will make frequent use of operations over both the real field $\mathbb{R}$ and the finite field consisting of the integers modulo p, $\mathbb{Z}_p = \{0, 1, \dots, p-1\}$, where p is prime.[4] Addition and summation over $\mathbb{R}$ will be denoted by $+$ and $\sum$, respectively. Similarly, addition and summation over the finite field $\mathbb{Z}_p$ where p is prime will be denoted by $\oplus$ and $\bigoplus$, respectively. We will write the modulo-p reduction of an integer $a \in \mathbb{Z}$ as $[a] \bmod p = r$ where $r \in \mathbb{Z}_p$ is the unique element satisfying $a = qp+r$ for some integer q. It will also be convenient to write the elementwise modulo-p reduction of an integer vector $\mathbf{a} \in \mathbb{Z}^L$ and an integer matrix $\mathbf{A} \in \mathbb{Z}^{M\times L}$ as $[\mathbf{a}] \bmod p$ and $[\mathbf{A}] \bmod p$, respectively. Recall that addition and multiplication over $\mathbb{Z}_p$ are equivalent to addition and multiplication over $\mathbb{R}$ followed by a modulo-p reduction, i.e.,

$$q_1 w_1 \oplus q_2 w_2 = [q_1 w_1 + q_2 w_2] \bmod p$$

where $q_1, q_2, w_1, w_2 \in \mathbb{Z}_p$. Note that on the right-hand side of the equation above, we have implicitly viewed q_1, q_2, w_1, w_2 as the corresponding elements of $\mathbb{Z}$ (under the natural mapping) in order to evaluate the real addition and multiplication. This will be the case throughout the paper: whenever elements of $\mathbb{Z}_p$ appear as part of operations over the reals, they will be implicitly viewed as the corresponding elements of $\mathbb{Z}$.

B. Channel Model

Consider L single-antenna transmitters that communicate to a receiver over a Gaussian multiple-access channel. See Figure 1 for an illustration. Each transmitter (indexed by

[1] Following the conference publication of this work, Tan *et al.* [66] noted that our proposed message representation may be inefficient for multi-hop relaying if each relay naively treats a linear combination over $\mathbb{Z}_p^k$ as k information symbols for the next hop. They proposed a lattice-based solution to this issue. It is also possible to resolve this issue directly over the message representation by having each relay only use some of its k received symbols as information symbols for the next hop. See [67, Section III.F] for details.

[2] Unlike [13], we do not propose a deterministic model for analyzing communication networks. Instead, here, we use a deterministic model as an expository tool to explain the decoding requirements of our problem statement.

[3] For the case of Eisenstein integers, our lattice achievability proof from Theorem 8 will need to be generalized following the approach of Huang *et al.* [27].

[4] Historically, the set of integers modulo p has been denoted by $\mathbb{Z}/p\mathbb{Z}$, $\mathbb{Z}/(p)$, or $\mathbb{Z}_p$. Some mathematicians prefer to avoid the notation $\mathbb{Z}_p$ because it can be confused with the set of p-adic integers if p is a prime number. Here, we will use the notation $\mathbb{Z}_p$ for the sake of conciseness, especially since we will frequently refer to vector spaces of the form $\mathbb{Z}_p^k$ (and have no need to refer to p-adic integers).

$\ell = 1, 2, \ldots, L$) produces a length-n *channel input* $\mathbf{x}_\ell \in \mathbb{R}^n$ subject to the *power constraint*[5]

$$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP_\ell \tag{1}$$

where $P_\ell \geq 0$.

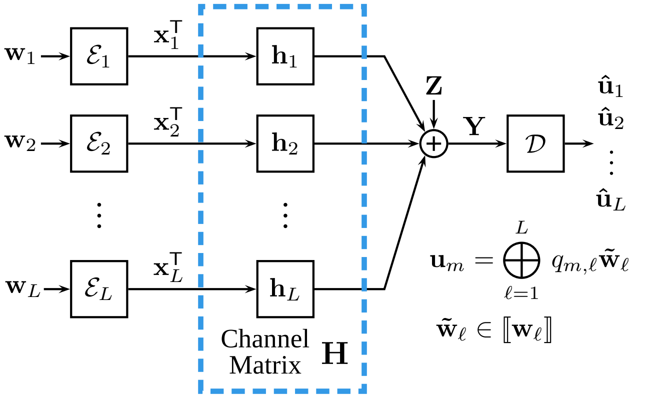
$\mathbf{w}_1$
$\mathcal{E}_1$
$\mathbf{x}_1^\mathsf{T}$
$\mathbf{h}_1$
$\mathbf{w}_2$
$\mathcal{E}_2$
$\mathbf{x}_2^\mathsf{T}$
$\mathbf{h}_2$
$\mathbf{w}_L$
$\mathcal{E}_L$
$\mathbf{x}_L^\mathsf{T}$
$\mathbf{h}_L$
$\mathbf{Z}$
$\mathbf{Y}$
$\mathcal{D}$
$\hat{\mathbf{u}}_1$
$\hat{\mathbf{u}}_2$
$\hat{\mathbf{u}}_L$
$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m,\ell}\tilde{\mathbf{w}}_\ell$
$\tilde{\mathbf{w}}_\ell \in [\![\mathbf{w}_\ell]\!]$
Channel Matrix $\mathbf{H}$

Fig. 1. Block diagram for the compute-and-forward problem with a single receiver. Each transmitter has a message $\mathbf{w}_\ell$ whose elements are taken from $\mathbb{Z}_p$. This message is embedded into $\mathbb{Z}_p^k$ (by zero-padding), mapped into a codeword $\mathbf{x}_\ell \in \mathbb{R}^n$, and sent over the channel. The receiver observes a noisy linear combination of these codewords, $\mathbf{Y} = \sum_\ell \mathbf{h}_\ell \mathbf{x}_\ell^\mathsf{T} + \mathbf{Z}$ and attempts to recover the linear combinations $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L$ of the coset representatives of the original messages.

The receiver has N_r antennas and observes an $N_\text{r} \times n$ dimensional *channel output* $\mathbf{Y}$ that is a noisy linear combination of the inputs:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{h}_\ell \mathbf{x}_\ell^\mathsf{T} + \mathbf{Z}$$

where $\mathbf{h}_\ell \in \mathbb{R}^{N_\text{r}}$ is the *channel vector* between the ℓth transmitter and the receiver and $\mathbf{Z} \in \mathbb{R}^{N_\text{r} \times n}$ is elementwise i.i.d. $\mathcal{N}(0, 1)$. It will often be convenient to group the channel vectors into a *channel matrix*

$$\mathbf{H} \triangleq \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_L \end{bmatrix} ,$$

and concisely write the channel output as

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \tag{2}$$

where $\mathbf{X} \triangleq \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_L \end{bmatrix}^\mathsf{T}$ is the matrix of channel inputs. We will assume throughout that the channel matrix $\mathbf{H}$ is known to the receiver and unknown to the transmitters. However, the transmitters may assume that the maximum singular value $\lambda_\text{max}(\mathbf{H})$ of the channel matrix is upper bounded by a constant.[6]

C. High-Level Overview

We now provide a high-level, informal overview of our compute-and-forward framework, which will help build intuition for the formal problem statement to follow. We begin by summarizing the original compute-and-forward framework from [1] and its MIMO generalization from [3]. We then discuss how to incorporate unequal power constraints and recovering multiple linear combinations into an expanded framework.

Compute-and-Forward with Equal Powers: The ℓth transmitter's message is a length-k_ℓ vector $\mathbf{w}_\ell$ whose elements are from $\mathbb{Z}_p$ where p is prime. Each message is zero-padded to a common length

$$\bar{\mathbf{w}}_\ell = \begin{bmatrix} \mathbf{w}_\ell \\ \mathbf{0}_{k - k_\ell} \end{bmatrix}$$

where $k = \max_\ell k_\ell$, and mapped to a length-n real-valued codeword $\mathbf{x}_\ell$ that satisfies the symmetric power constraint $\mathbb{E}\big[\|\mathbf{x}_\ell\|^2\big] \leq nP$. The rate R_ℓ associated with a transmitter is the number of bits in its message normalized by the length of the codeword, $R_\ell = (k_\ell/n)\log p$. Given coefficients $q_1, q_2, \ldots, q_L \in \mathbb{Z}_p$, the receiver's goal is to recover a linear combination $\mathbf{u}$ of the (zero-padded) messages,

$$\mathbf{u} = \bigoplus_{\ell=1}^{L} q_\ell \bar{\mathbf{w}}_\ell .$$

As argued in [1], the main idea underlying compute-and-forward is to establish a connection between linear combinations of the messages and *integer-linear combinations of the codewords*, in order to exploit the noisy linear combination taken by the channel. For instance, after applying an equalization vector $\mathbf{b} \in \mathbb{R}^{N_\text{r}}$, the channel output can be expressed as an integer-linear combination of the codewords[7] with coefficients $a_1, a_2, \ldots, a_L \in \mathbb{Z}$ plus effective noise,

$$\mathbf{b}^\mathsf{T}\mathbf{Y} = \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell^\mathsf{T} + \underbrace{\sum_{\ell=1}^{L} \big(\mathbf{b}^\mathsf{T}\mathbf{h}_\ell - a_\ell\big)\mathbf{x}_\ell^\mathsf{T} + \mathbf{b}^\mathsf{T}\mathbf{Z}}_{\text{effective noise}} . \tag{3}$$

Each integer-linear combination of codewords is associated with a linear combination of the messages with coefficients $q_\ell = [a_\ell] \bmod p$. The performance of a compute-and-forward scheme is given by a computation rate region, which is specified by a function $R_\text{comp}(\mathbf{H}, \mathbf{a})$ that maps each channel matrix $\mathbf{H}$ and integer coefficient vector $\mathbf{a} = [a_1 \ a_2 \ \cdots \ a_L]^\mathsf{T}$ to a rate. Specifically, if the rates associated to messages with non-zero coefficients are less than the computation rate

$$\max_{\ell : a_\ell \neq 0} R_\ell < R_\text{comp}(\mathbf{H}, \mathbf{a}) ,$$

then the linear combination with coefficients $q_\ell = [a_\ell] \bmod p$ is decodable with vanishing probability of error (with respect to the blocklength n). Operationally, this means that the scheme works in the absence of channel state information at the transmitter (CSIT) and that the receiver is free to

[5] In [1, Appendix C], it is argued that, for symmetric compute-and-forward, the expected power constraint $\mathbb{E}\big[\|\mathbf{x}_\ell\|^2\big] \leq nP$ can be replaced with a hard power constraint $\|\mathbf{x}\|^2 \leq nP$ without affecting the achievable rates. A similar argument should apply in our setting by first refining the nested lattice existence proof in [59, Theorem 2] to show that the coarse lattices are also good for covering [58]. Alternatively, each encoder can throw out a constant fraction of its codebook to obtain a subcodebook that satisfies a hard power constraint while still maintaining the same achievable rate asymptotically.

[6] This assumption will be used to make a connection between the solvability of the linear combinations over $\mathbb{Z}_p$ and the rank of the integer matrix $\mathbf{A}$ over $\mathbb{R}$. If we further assume that the channel matrix is generated randomly and the receiver is able to tolerate some probability of outage, then this condition can by replaced by the milder condition that $\mathbb{P}\big(\lambda_\text{max}(\mathbf{H}) \geq \gamma\big) \to 0$ as $\gamma \to \infty$. See [1, Remark 10] for further details.

[7] To be precise, our coding scheme employs dithered lattice codewords as the channel inputs $\mathbf{x}_\ell$. However, the dithers can be removed at the receiver prior to decoding, and are thus ignored in this high-level overview.

choose which linear combination to decode, among those satisfying the computation rate constraint. Owing to this form of universality, compute-and-forward is applicable to scenarios with multiple receivers, each facing a different channel matrix and aiming to decode its own linear combination.

It shown in [1, Theorem 1] and [3, Theorem 3] that the computation rate region described by

$$R_{\text{comp}}(\mathbf{H},\mathbf{a}) = \frac{1}{2}\log^+\left(\frac{P}{\mathbf{a}^\mathsf{T}\left(P^{-1}\mathbf{I}+\mathbf{H}^\mathsf{T}\mathbf{H}\right)^{-1}\mathbf{a}}\right) \tag{4}$$

is achievable. The achievability proof utilizes nested lattice codebooks, which guarantees that any integer-linear combination of codewords is itself a codeword and thus afforded protection from noise. In particular, each transmitter's codebook is constructed using a fine lattice with effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and a common coarse lattice that enforces the power constraint P. These nested lattices are chosen such that the ℓth transmitter's rate R_ℓ converges to $\frac{1}{2}\log^+(P/\sigma^2_{\text{eff},\ell})$ asymptotically in the blocklength n. The nested lattice construction sends the field size p to infinity with the blocklength n, in order to produce Gaussian-like channel inputs and obtain closed-form rate expressions.

Remark 1: If the field size is held fixed, we encounter similar issues as seen when evaluating the capacity of a point-to-point Gaussian channel under a finite input alphabet, i.e., we do not obtain closed-form rate expressions. For practical point-to-point codes, a common approach is to pick a finite constellation size based on the SNR [69] and accept a small rate loss. A similar approach enables practical codes for compute-and-forward [14], [31]. ◊

Remark 2: Since the field size p changes with the blocklength n, it does not make sense to specify the desired linear combinations via fixed coefficients $q_\ell \in \mathbb{Z}_p$. Instead, we fix desired integer coefficients a_ℓ and specify the desired linear combinations as those with coefficients satisfying $q_{m,\ell} = [a_{m,\ell}] \bmod p$. ◊

The effective noise tolerance of the codeword associated to $\sum_\ell a_\ell \mathbf{x}_\ell$ is determined by the minimum noise tolerance over all participating fine lattices, $\min_{\ell: a_\ell \neq 0} \sigma^2_{\text{eff},\ell}$. Roughly speaking, an integer-linear combination is decodable if the variance of the effective noise in (3) is less than its effective noise tolerance. It can be shown that the denominator in (4) corresponds to the variance of the effective noise when $\mathbf{b}$ is chosen as the minimum mean-squared error (MMSE) projection.

Compute-and-Forward with Unequal Powers: In this paper, we expand the original compute-and-forward framework [1] in two aspects. First, we allow for an unequal power allocation across transmitters. Second, we explicitly consider the scenario where the receiver may wish to recover more than one linear combination. Decoding more than one linear combination appears in many contexts, such as recovering the L transmitted messages in an integer-forcing MIMO receiver [3], relaying in a network where there are more transmitters than relays, and integer-forcing interference alignment [10]. In order to incorporate these two generalizations, we will expand the definition of the computation rate region. Here, we provide an intuitive description of our modifications before presenting a formal problem statement.

We first describe our modification to the message structure. To each transmitter, we associate a power constraint P_ℓ and, as before, an effective noise tolerance $\sigma^2_{\text{eff},\ell}$. In the equal power setting, the rates varied across transmitters due only to the change in the effective noise tolerance. To cope with the fact that the messages have different lengths, they are zero-padded prior to taking linear combinations. Here, the rates will vary due to both changes in power and effective noise tolerance, for which zero-padding will not suffice. Instead, we take inspiration from the idea of *signal levels* as introduced in [13].

The length of the ℓth transmitter's message is a length-$(k_{\text{F},\ell} - k_{\text{C},\ell})$ vector $\mathbf{w}_\ell$ whose elements are drawn from $\mathbb{Z}_p$ where p is prime and the parameters $k_{\text{C},\ell}, k_{\text{F},\ell} \in \mathbb{N}$ will be determined by the power constraint P_ℓ and effective noise tolerance $\sigma^2_{\text{eff},\ell}$, respectively. Define $k_\text{C} = \min_\ell k_{\text{C},\ell}$ and $k_\text{F} = \max_\ell k_{\text{F},\ell}$. The total number of available signal levels is $k = k_\text{F} - k_\text{C}$. Each message is embedded into $\mathbb{Z}_p^k$ using $k_{\text{C},\ell} - k_\text{C}$ leading zeros and $k_\text{F} - k_{\text{F},\ell}$ trailing zeros,

$$\begin{bmatrix} \mathbf{0}_{k_{\text{C},\ell}-k_\text{C}} \\ \mathbf{w}_\ell \\ \mathbf{0}_{k_\text{F}-k_{\text{F},\ell}} \end{bmatrix}, \tag{5}$$

and mapped to a length-n real-valued codeword $\mathbf{x}_\ell$ that satisfies the power constraint $\mathbb{E}\left[\|\mathbf{x}_\ell\|^2\right] \leq nP_\ell$. The rate R_ℓ associated with this transmitter is the number of message bits normalized by the codeword length, $R_\ell = ((k_{\text{F},\ell} - k_{\text{C},\ell})/n)\log p$.

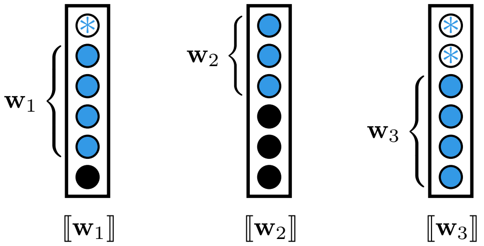

Fig. 2. Illustration of message cosets for $L = 3$ transmitters. The parameters are $k_{\text{F},1} = 7$, $k_{\text{F},2} = 5$, $k_{\text{F},3} = 8$, $k_{\text{C},1} = 3$, $k_{\text{C},2} = 2$, and $k_{\text{C},3} = 4$. Therefore, $k_\text{F} = 8$, $k_\text{C} = 2$, and $k = 6$. We use the symbol $*$ to stand for a "don't care" entry that can take any value in $\mathbb{Z}_p$, lightly shaded (blue) circles to denote information symbols (i.e., elements of $\mathbf{w}_\ell$) that can take values in $\mathbb{Z}_p$, and black circles to denote zeros. Based on the parameters, the coset $[\![\mathbf{w}_1]\!]$ associated with the first transmitter has 1 "don't care" entry, then 4 information symbols, and then 1 zero. Similarly, the coset $[\![\mathbf{w}_2]\!]$ associated with the second transmitter has 3 information symbols followed by 3 zeros. Finally, the coset $[\![\mathbf{w}_3]\!]$ associated with the third transmitter has 2 "don't care" entries followed by 4 information symbols. The receiver's goal is to recover a linear combination of vectors drawn from these cosets.

As mentioned above, the receiver may wish to decode more than one linear combination. We compactly represent the receiver's demands through L desired linear combinations $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L$ of the form

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m,\ell}\tilde{\mathbf{w}}_\ell$$

where $q_{m,\ell} \in \mathbb{Z}_p$ and $\tilde{\mathbf{w}}_\ell$ is an element of a certain coset

$[\![\mathbf{w}_\ell]\!]$ with respect to the message $\mathbf{w}_\ell$.[8] Specifically, the coset consists of all vectors in $\mathbb{Z}_p^k$ for which the first $k_{\text{C},\ell} - k_{\text{C}}$ elements can take any values in $\mathbb{Z}_p$ (and can be viewed as "don't care" entries), the next $k_{\text{F},\ell} - k_{\text{C},\ell}$ elements contain the message $\mathbf{w}_\ell$, and the remaining $k_{\text{F}} - k_{\text{F},\ell}$ entries are equal to zero.

It is convenient to group the coefficients into a matrix $\mathbf{Q} = \{q_{m,\ell}\}$. If the receiver wants fewer than L linear combinations, then it can set the entries of the unneeded rows of $\mathbf{Q}$ to zero. We have illustrated an example message structure in Figure 2. Note that this framework includes the original problem statement as a special case by setting $k_{\text{C},1} = k_{\text{C},2} = \ldots = k_{\text{C},L}$.

We have relaxed the decoding requirements by allowing the receiver to decode linear combinations of vectors drawn from the same cosets as the message vectors. However, this does not affect the algebraic conditions for recovering messages from their linear combinations. Specifically, if the coefficient matrix $\mathbf{Q}$ enables the receiver to recover $\tilde{\mathbf{w}}_\ell$, it can also immediately recover $\mathbf{w}_\ell$. For example, if $\mathbf{Q}$ is full rank, the receiver can recover all L transmitted messages.

Our coding scheme will employ nested lattice codes in order to link linear combinations of the messages to integer-linear combinations of the codewords, just as in the symmetric case. We will select $k_{\text{C},\ell}$ using the transmitter's power constraint P_ℓ and $k_{\text{F},\ell}$ using the effective noise tolerance $\sigma^2_{\text{eff},\ell}$ so that the ℓth transmitter's rate R_ℓ converges to $\frac{1}{2}\log^+(P_\ell/\sigma^2_{\text{eff},\ell})$ asymptotically in the blocklength n. As before, the field size p tends to infinity with the blocklength n. Thus, we select desired integer coefficients $a_{m,\ell} \in \mathbb{Z}$ and specify the desired linear combinations as those with coefficients satisfying $q_{m,\ell} = [a_{m,\ell}] \bmod p$.

The channel output can be written as an integer-linear combination of the codewords plus effective noise as in (3). We consider L such integer-linear combinations and collect the desired coefficients into an integer coefficient matrix $\mathbf{A} = \{a_{m,\ell}\}$, which in turn specifies the coefficient matrix as $\mathbf{Q} = [\mathbf{A}] \bmod p$ where the modulo operation is taken elementwise.

We now describe our modification to the computation rate region definition from [1]. Unlike the equal power setting, the rate region cannot be described using a single computation rate. For example, even if we are interested in only recovering a single linear combination at the receiver, the computation rate for each transmitter will still be determined by its own power combined with the effective noise for the linear combination. As another example, consider an equal power setting where the receiver wishes to decode more than one linear combination. Once it has decoded a single linear combination, it can use it as side information to help decode the next, meaning that the rate constraints for the linear combinations should be considered jointly. To capture such phenomena, we characterize the performance of a compute-and-forward scheme via a set-valued computation rate function $\mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A})$ that maps each channel matrix $\mathbf{H}$ and integer coefficient matrix $\mathbf{A}$ to a subset of $\mathbb{R}_+^L$. This subset consists of all rate tuples that are achievable for the specified $\mathbf{H}$ and $\mathbf{A}$ under the chosen coding scheme. That is, if the rate tuple associated to the messages falls inside the computation rate region,

$$(R_1, R_2, \ldots, R_L) \in \mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A}) \ ,$$

then the linear combinations with coefficient matrix $\mathbf{Q} = [\mathbf{A}] \bmod p$ are decodable with vanishing probability of error (with respect to the blocklength n). See Figure 3 for an illustration explaining the computation rate region.

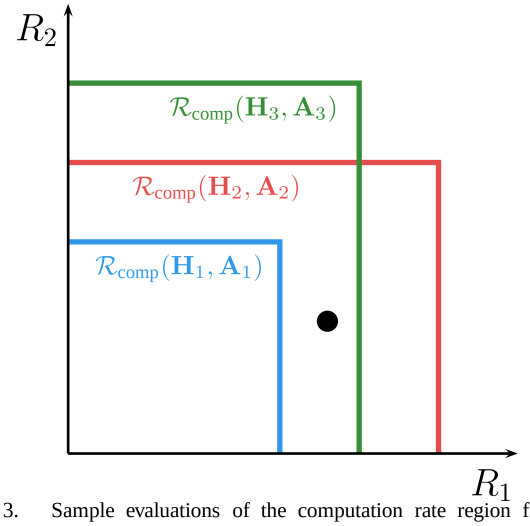
R_2
R_1
$\mathcal{R}_{\text{comp}}(\mathbf{H}_3, \mathbf{A}_3)$
$\mathcal{R}_{\text{comp}}(\mathbf{H}_2, \mathbf{A}_2)$
$\mathcal{R}_{\text{comp}}(\mathbf{H}_1, \mathbf{A}_1)$

Fig. 3. Sample evaluations of the computation rate region for $L = 2$ transmitters. The dot denotes the rate tuple for the two transmitters. Since this rate tuple falls inside the (red) rate region $\mathcal{R}_{\text{comp}}(\mathbf{H}_2, \mathbf{A}_2)$, the receiver can recover the linear combinations with integer coefficient matrix $\mathbf{A}_2$ under channel matrix $\mathbf{H}_2$. Similarly, the linear combinations with integer coefficient matrix $\mathbf{A}_3$ can be recovered under channel matrix $\mathbf{H}_3$ since the rate tuple falls inside the (green) rate region $\mathcal{R}_{\text{comp}}(\mathbf{H}_3, \mathbf{A}_3)$. On the other hand, since the rate tuple falls outside the (blue) rate region $\mathcal{R}_{\text{comp}}(\mathbf{H}_1, \mathbf{A}_1)$, the receiver is not required to recover the linear combinations with integer coefficient matrix $\mathbf{A}_1$ under channel matrix $\mathbf{H}_1$.

D. Formal Problem Statement

We now provide a formal problem statement. A coding scheme is parametrized by the following:

- A positive integer n denoting coding blocklength,
- a positive prime number p denoting the size of the finite field $\mathbb{Z}_p$ over which the linear combinations are taken, and
- non-negative integers $k_{\text{C},\ell}, k_{\text{F},\ell}$ for $\ell \in \{1, 2, \ldots, L\}$ satisfying $k_{\text{C},\ell} \leq k_{\text{F},\ell} \leq n$, where $k_{\text{F},\ell} - k_{\text{C},\ell}$ can be interpreted as the number of available signal levels at the ℓth transmitter.

Definition 1 (Messages): For $\ell = 1, 2, \ldots, L$, the ℓth transmitter has a *message* $\mathbf{w}_\ell$ that is drawn independently and uniformly over $\mathbb{Z}_p^{k_{\text{F},\ell} - k_{\text{C},\ell}}$. The *rate* of the ℓth message (in bits per channel use) is

$$\frac{k_{\text{F},\ell} - k_{\text{C},\ell}}{n} \log p \ .$$

◇

Definition 2 (Encoders): For $\ell = 1, 2, \ldots, L$, the ℓth transmitter is equipped with an *encoder* $\mathcal{E}_\ell : \mathbb{Z}_p^{k_{\text{F},\ell} - k_{\text{C},\ell}} \rightarrow \mathbb{R}^n$ that

[8] To be precise, the linear combinations are affine varieties (i.e., translates of a vector subspace). However, we will simply refer to these as linear combinations throughout the paper.

maps its message into a *channel input vector* $\mathbf{x}_\ell = \mathcal{E}_\ell(\mathbf{w}_\ell)$ subject to the power constraint (1). $\Diamond$

Definition 3 (Decoder): Define $k_{\text{C}} \triangleq \min_\ell k_{\text{C},\ell}$, $k_{\text{F}} \triangleq \max_\ell k_{\text{F},\ell}$, and $k \triangleq k_{\text{F}} - k_{\text{C}}$. Also, define the coset

$$[\![\mathbf{w}_\ell]\!] \triangleq \left\{ \mathbf{w} \in \mathbb{Z}_p^k : \mathbf{w} = \begin{bmatrix} \mathbf{r} \\ \mathbf{w}_\ell \\ \mathbf{0}_{k_{\text{F}} - k_{\text{F},\ell}} \end{bmatrix} \text{ for some } \mathbf{r} \in \mathbb{Z}_p^{k_{\text{C},\ell} - k_{\text{C}}} \right\} \tag{6}$$

The receiver is equipped with a *decoder* $\mathcal{D} : \mathbb{R}^{N_{\text{r}} \times n} \times \mathbb{R}^{N_{\text{r}} \times L} \times \mathbb{Z}^{L \times L} \to \mathbb{Z}_p^{L \times k}$ that takes as inputs the channel observation $\mathbf{Y}$ from (2), the channel matrix $\mathbf{H}$, and the desired integer coefficient matrix $\mathbf{A}$, and outputs an estimate $\hat{\mathbf{U}} = \mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{A})$. Let $\hat{\mathbf{u}}_m^{\mathsf{T}}$ denote the mth row of $\mathbf{U}$. We say that decoding is *successful* if $\hat{\mathbf{u}}_1 = \mathbf{u}_1, \hat{\mathbf{u}}_2 = \mathbf{u}_2, \ldots, \hat{\mathbf{u}}_L = \mathbf{u}_L$ for some linear combinations of the form

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m,\ell} \tilde{\mathbf{w}}_\ell$$

where the $q_{m,\ell} \in \mathbb{Z}_p$ are the entries of $\mathbf{Q} = [\mathbf{A}] \bmod p$ and $\tilde{\mathbf{w}}_\ell \in [\![\mathbf{w}_\ell]\!]$. We say that the decoder makes an *error* if it is not successful. We sometimes refer to $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L$ as linear combinations with integer coefficient matrix $\mathbf{A}$. $\Diamond$

Definition 4 (Computation Rate Region): A *computation rate region* is specified by a set-valued function $\mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A})$ that maps each channel matrix $\mathbf{H} \in \mathbb{R}^{N_r \times L}$ and integer coefficient matrix $\mathbf{A} \in \mathbb{Z}^{L \times L}$ to a subset of $\mathbb{R}_+^L$. The computation rate region described by a set-valued function $\mathcal{R}_{\text{comp}}$ is *achievable* if, for every rate tuple $(R_1, R_2, \ldots, R_L) \in \mathbb{R}_+^L$, $\epsilon > 0$, and n large enough, there exist

- parameters $p, k_{\text{C},\ell}, k_{\text{F},\ell}$ satisfying $\dfrac{k_{\text{F},\ell} - k_{\text{C},\ell}}{n} \log p > R_\ell - \epsilon$ for $\ell = 1, 2, \ldots, L$,
- encoders $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_L$,

such that,

- for every channel matrix $\mathbf{H} \in \mathbb{R}^{N_r \times L}$ and
- every integer matrix $\mathbf{A}$, satisfying $(R_1, R_2, \ldots, R_L) \in \mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A})$,

there exists a decoder $\mathcal{D}$ with probability of decoding error at most ϵ. $\Diamond$

Remark 3: The usual approach to defining a rate region is to first define the notion of an achievable rate tuple, and then define the rate region as the set of all achievable rate tuples. Our definition does not have this structure because the encoders $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_L$ are assumed to be ignorant of both the channel matrix $\mathbf{H}$ and the integer coefficient matrix $\mathbf{A}$. Thus, a rate tuple $(R_1, R_2, \ldots, R_L)$ selected by the encoders will not lead to successful decoding for all $(\mathbf{H}, \mathbf{A})$ pairs. Instead, we characterize the rate region via the set-valued function $\mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A})$, which specifies the rate tuples that lead to successful decoding for each $(\mathbf{H}, \mathbf{A})$ pair. $\Diamond$

Remark 4: In some cases, it may be possible to simplify the framework by setting all of the "don't care" entries to zero, i.e., setting $\mathbf{r} = \mathbf{0}$ in (6). For example, if the encoders do not dither their lattice codewords prior to transmission in the proof of Theorem 1, then it follows from Definition 11 that this is possible. However, this will significantly complicate the proof, since the effective noise will not be independent from the desired linear combination. More generally, this problem statement is directly applicable for compute-and-forward over discrete memoryless networks [18]. In that setting, the "don't care" entries play an important role for selecting codewords with the desired type, following the joint typicality encoding approach of Padakandla and Pradhan [70]. $\Diamond$

III. Main Results

In this section, we state our main coding theorems as well as provide intuitions and examples. Although the proofs of our achievability results rely on the existence of good nested lattice codes, we have deferred (nearly all) discussion of lattices to subsequent sections in order to make our main results more accessible. Our primary technical contribution is the generalization of the compute-and-forward framework to allow for unequal powers, multiple receive antennas, and recovering more than one linear combination (at a single receiver), all while maintaining a connection to $\mathbb{Z}_p^k$. We demonstrate the utility of our generalization of the compute-and-forward framework by applying it to the classical and compound Gaussian multiple-access channels.

Note that all of our results implicitly assume that the transmitters do not have access to channel state information. It is well-known that, with channel state information, the transmitters can steer the channel gains towards integer values, which can improve the end-to-end rates [41], [71]. This can be captured within our framework by multiplying each channel gain by a scalar (chosen using channel state information). More generally, the achievable rates of linear beamforming strategies for multi-antenna transmitters can be captured within our framework by multiplying each transmitter's channel vector (on the right) by its beamforming vector. See [10] for an application to interference alignment and [17] for an application to uplink-downlink duality.

We now provide a high-level summary of our results:

- **Parallel Computation:** Theorem 1 expands the computation rate region from [1] by permitting unequal power allocation across the transmitters. Each of the desired linear combinations is decoded independently of the others, which we refer to as "parallel computation."
- **Successive Computation:** Theorem 2 enlarges the computation rate region from Theorem 1 by decoding the linear combinations one-by-one and employing successive cancellation. This "successive computation" technique can be viewed as a generalization of [7], [16] to the unequal power setting. Theorem 3 shows that it suffices to use "primitive" integer coefficient matrices, i.e., integer matrices with a unimodular completion.
- **Multiple-Access Sum Capacity within a Constant Gap:** Theorem 4 shows that the sum of the L highest parallel computation rates always lies within L bits of the sum capacity of the underlying multiple-access channel. Furthermore, these computation rates can mapped to individual users, which leads to an operational interpretation as a multiple-access strategy. An implication of the theorem is that the parallel computation strategy,

when combined with algebraic successive cancellation, is approximately optimal for multiple access. From one perspective, Theorem 4 generalizes the compute-and-forward transform of [9] to the unequal power setting.

- **Multiple-Access Sum Capacity:** Theorem 5 shows that, for any unimodular integer coefficient matrix, the sum of the L successive computation rates is exactly equal to the sum capacity of the underlying multiple-access channel. Under certain technical conditions, this can be employed as an optimal multiple-access strategy. Theorem 5 generalizes the successive integer-forcing scheme of [7] to the unequal power setting.
- **Multiple Receivers:** Theorems 6 and 7 give achievable rate regions for multiple receivers for parallel and successive computation, respectively.

We now introduce some additional notation. Define $\mathbf{P}$ to be the diagonal matrix of the power constraints,

$$\mathbf{P} \triangleq \text{diag}(P_1, P_2, \dots, P_L) \ ,$$

and let $\mathbf{F} \in \mathbb{R}^{L \times L}$ be any matrix that satisfies

$$\mathbf{F}^{\mathsf{T}}\mathbf{F} = \left(\mathbf{P}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1} \ . \tag{7}$$

Note that $\mathbf{F}$ is not unique and can determined via several approaches, such as via its eigendecomposition or its Cholesky decomposition.

Recall that $\mathbf{A} \in \mathbb{Z}^{L \times L}$ is the desired integer coefficient matrix. Let $\mathbf{a}_m^{\mathsf{T}}$ denote the mth row of $\mathbf{A}$ and $a_{m,\ell}$ denote the (m,ℓ)th entry. We will sometimes refer to $\mathbf{u}_m$ as the *linear combination with integer coefficient vector* $\mathbf{a}_m$. In certain scenarios (e.g., relaying, interference alignment), the receiver may wish to decode $M < L$ linear combinations. This can be explicitly represented in our framework by setting the last $M - L$ rows of $\mathbf{A}$ to be zero but it will be convenient to develop more compact notation. Let $\bar{\mathbf{A}}$ be an $M \times L$ integer matrix with $M < L$. We will implicitly take $\mathcal{R}_{\text{comp}}(\mathbf{H}, \bar{\mathbf{A}})$ to mean $\mathcal{R}_{\text{comp}}(\mathbf{H}, \mathbf{A})$ where

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{O}_{(L-M)\times L} \end{bmatrix} \ .$$

We also recall the following basic result from linear algebra.

Lemma 1 (Woodbury Matrix Identity): For any (appropriately-sized) matrices $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$, and $\mathbf{M}_4$, we have that

$$\begin{aligned}&\left(\mathbf{M}_1 + \mathbf{M}_2\mathbf{M}_3\mathbf{M}_4\right)^{-1} \\ &= \mathbf{M}_1^{-1} - \mathbf{M}_1^{-1}\mathbf{M}_2\left(\mathbf{M}_3^{-1} + \mathbf{M}_4\mathbf{M}_1^{-1}\mathbf{M}_2\right)^{-1}\mathbf{M}_4\mathbf{M}_1^{-1}\end{aligned}$$

□

See, e.g., [72, Theorem 18.2.8] for a proof. As an example, take $\mathbf{M}_1 = \mathbf{P}^{-1}$, $\mathbf{M}_2 = \mathbf{H}^{\mathsf{T}}$, $\mathbf{M}_3 = \mathbf{I}$, and $\mathbf{M}_4 = \mathbf{H}$. It then follows from the Woodbury Identity that

$$\left(\mathbf{P}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1} = \mathbf{P} - \mathbf{P}\mathbf{H}^{\mathsf{T}}\left(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\right)^{-1}\mathbf{H}\mathbf{P} \ . \tag{8}$$

We will make frequent use of this identity.

A. Parallel Computation

We begin with a "parallel computation" strategy, in which the receiver decodes each of the desired linear combinations independently. To recover the mth linear combination, the receiver applies an equalization vector $\mathbf{b}_m \in \mathbb{R}^{N_{\text{r}}}$ to its observation to obtain the effective channel

$$\begin{aligned}\tilde{\mathbf{y}}_m &= \mathbf{b}_m^{\mathsf{T}}\mathbf{Y} \\ &= \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_m^{\mathsf{T}}\mathbf{X} \ + \ \underbrace{\left(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}}\right)\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}}_{\text{effective noise}} \ ,\end{aligned}$$

and then decodes to the closest lattice codeword. In Section V, we will argue that, if the fine lattices associated with $\mathbf{a}_m^{\mathsf{T}}\mathbf{X}$ can tolerate an effective noise variance of

$$\begin{aligned}\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m) &\triangleq \frac{1}{n}\mathbb{E}\left\|\left(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}}\right)\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}\right\|^2 \\ &= \|\mathbf{b}_m\|^2 + \left\|\left(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}}\right)\mathbf{P}^{1/2}\right\|^2 \ , \end{aligned} \tag{9}$$

then the linear combination with integer coefficient vector $\mathbf{a}_m$ can be successfully decoded.

Lemma 2: The equalization vector $\mathbf{b}_m \in \mathbb{R}^{N_{\text{r}}}$ that minimizes the effective noise variance from (9) is the MMSE projection vector

$$\mathbf{b}_{\text{opt},m}^{\mathsf{T}} = \mathbf{a}_m^{\mathsf{T}}\mathbf{P}\mathbf{H}^{\mathsf{T}}\left(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\right)^{-1} \ .$$

The minimal effective noise variance is

$$\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m) \triangleq \sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_{\text{opt},m}) \tag{10}$$

$$= \mathbf{a}_m^{\mathsf{T}}\left(\mathbf{P}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1}\mathbf{a}_m$$

$$= \|\mathbf{F}\mathbf{a}_m\|^2 \ , \tag{11}$$

where $\mathbf{F}$ is any matrix that satisfies (7). □

See Appendix A for a proof.

The users that participate in the integer-linear combination $\mathbf{a}_m^{\mathsf{T}}\mathbf{X}$ are simply those with non-zero integer coefficients, $a_{m,\ell} \neq 0$. These users should satisfy the rate constraints $R_\ell < 1/2 \log^+\left(P_\ell/\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m)\right)$ in order for the receiver to directly decode the linear combination. Note that the receiver can also decode linear combinations *indirectly*. Specifically, it can decode *any* integer-linear combinations whose integer coefficient matrix $\tilde{\mathbf{A}} \in \mathbb{Z}^{L \times L}$ has the same rowspan as $\mathbf{A}$, and then simply solve for $\mathbf{A}\mathbf{X}$ from $\tilde{\mathbf{A}}\mathbf{X}$. Therefore, the achievable computation rate region involves a union over all integer matrices with the same rowspan, as captured by the following theorem.

Theorem 1: For an AWGN network with L transmitters, a receiver, and power constraints $P_1, \dots, P_L$, the following computation rate region is achievable,

$$\mathcal{R}^{(\text{para})}_{\text{comp}}(\mathbf{H}, \mathbf{A}) = \bigcup_{\substack{\tilde{\mathbf{A}} \in \mathbb{Z}^{L \times L} \\ \text{rowspan}(\mathbf{A}) \subseteq \text{rowspan}(\tilde{\mathbf{A}})}} \mathcal{R}_{\text{para}}(\mathbf{H}, \tilde{\mathbf{A}})$$

where

$$\mathcal{R}_{\text{para}}(\mathbf{H}, \tilde{\mathbf{A}}) \triangleq \Bigg\{ (R_1, \dots, R_L) \in \mathbb{R}^L_+ \ :$$

$$R_\ell \leq \frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H}, \tilde{\mathbf{a}}_m)}\right) \ \forall (m,\ell) \text{ s.t. } \tilde{a}_{m,\ell} \neq 0 \Bigg\}$$

and $\tilde{\mathbf{a}}_m^{\mathsf{T}}$ and $\tilde{a}_{m,\ell}$ are the mth row and (m,ℓ)th entry of $\tilde{\mathbf{A}}$, respectively. □

The achievability proof is presented in Section V.

Remark 5: The computation rate region described in Theorem 1, when restricted to the special case of equal powers and a single antenna at the receiver, yields the rate region from [1, Theorem 1].[9] ♢

The following lemma restricts the search space for integer vectors.

Lemma 3: Let $\lambda_{\text{max}}\big(\mathbf{I} + \mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)$ denote the maximum eigenvalue of $\mathbf{I} + \mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}$. Consider an integer matrix $\tilde{\mathbf{A}} \in \mathbb{Z}^{L\times L}$ and user index $\ell \in \{1,2,\ldots,L\}$. If, for some $m \in \{1,2,\ldots,L\}$, the (m,ℓ)th entry of $\tilde{\mathbf{A}}$ satisfies

$$a^2_{m,\ell} > \lambda_{\text{max}}\big(\mathbf{I} + \mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big) \ ,$$

then $R_\ell = 0$ for any rate tuple $(R_1,\ldots,R_L) \in \mathcal{R}_{\text{para}}(\mathbf{H}, \tilde{\mathbf{A}})$. □

The proof is deferred to Appendix B.

Example 1: Consider a receiver that observes $Y = X_1 + \cdots + X_L + Z$ and wants to decode the linear combination with integer coefficient vector $\mathbf{a}_1 = \mathbf{1}$ (i.e., the sum of the messages over $\mathbb{Z}_p$). (For $L = 2$, this corresponds to the multiple-access phase in a Gaussian two-way relay channel as studied by [26], [73].) Using (8), we get

$$\begin{aligned}\sigma^2_{\text{para}}(\mathbf{1}^{\mathsf{T}}, \mathbf{1}^{\mathsf{T}}) &= \mathbf{1}^{\mathsf{T}}\mathbf{P}\mathbf{1} - \mathbf{1}^{\mathsf{T}}\mathbf{P}\mathbf{1}\big(1 + \mathbf{1}^{\mathsf{T}}\mathbf{P}\mathbf{1}\big)^{-1}\mathbf{1}^{\mathsf{T}}\mathbf{P}\mathbf{1} \\ &= \frac{\sum_{\ell=1}^{L} P_\ell}{1 + \sum_{\ell=1}^{L} P_\ell} \ .\end{aligned}$$

The resulting rate region is

$$\begin{aligned}&\mathcal{R}_{\text{para}}(\mathbf{1}^{\mathsf{T}}, \mathbf{1}^{\mathsf{T}}) \\ &= \left\{(R_1,\ldots,R_L) \in \mathbb{R}^L_+ \ : \ R_\ell \leq \frac{1}{2}\log^+\left(\frac{P_\ell}{\sum_{i=1}^{L} P_i} + P_\ell\right)\right\}\end{aligned}$$

and is equal to that derived by Nam, Chung, and Lee [12] for decoding the sum of lattice codewords over a multiple-access channel with unequal powers.

It is well-known that this region can be expanded at low SNR by decoding the messages individually and then computing the sum. That is, the rowspace of $\tilde{\mathbf{A}} = \mathbf{I}$ contains that of $\mathbf{A}$ and yields the rate region

$$\begin{aligned}&\mathcal{R}_{\text{para}}(\mathbf{1}^{\mathsf{T}}, \mathbf{I}) \\ &= \left\{(R_1,\ldots,R_L) \in \mathbb{R}^L_+ \ : \ R_\ell \leq \frac{1}{2}\log\left(1 + \frac{P_\ell}{\sum_{i\neq\ell} P_i}\right)\right\} .\end{aligned}$$

The computation rate region can be further expanded by taking the union over all viable $\tilde{\mathbf{A}}$, e.g., by first decoding sums of subsets of the messages and then combining these. ♢

Example 2: Consider a receiver that wishes to recover all of the messages, $\mathbf{A} = \mathbf{I}$. Let $\boldsymbol{\delta}_m$ denote the mth column of $\mathbf{I}$. Using (8), the effective noise variances are

$$\begin{aligned}\sigma^2_{\text{para}}(\mathbf{H}, \boldsymbol{\delta}_m) &= \boldsymbol{\delta}_m^{\mathsf{T}}\big(\mathbf{P} - \mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1}\mathbf{H}\mathbf{P}\big)\boldsymbol{\delta}_m \\ &= P_m - P_m^2\mathbf{h}_m^{\mathsf{T}}\big(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1}\mathbf{h}_m \ .\end{aligned}$$

The effective SNR of the ℓth user is

$$\begin{aligned}\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H}, \boldsymbol{\delta}_\ell)} &= \frac{1}{1 - P_\ell\mathbf{h}_\ell^{\mathsf{T}}\big(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1}\mathbf{h}_\ell} \\ &= 1 + P_\ell\mathbf{h}_\ell^{\mathsf{T}}\Big(\mathbf{I} + \sum_{i\neq\ell} P_i\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}\Big)^{-1}\mathbf{h}_\ell\end{aligned}$$

where the last step uses the Woodbury Matrix Identity (Lemma 1) with $\mathbf{M}_1 = 1$, $\mathbf{M}_2 = P_\ell^{1/2}\mathbf{h}_\ell^{\mathsf{T}}$, $\mathbf{M}_3 = \big(\mathbf{I} + \sum_{i\neq\ell} P_i\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}\big)^{-1}$, and $\mathbf{M}_4 = \mathbf{M}_2^{\mathsf{T}}$. Finally, the rate region (for direct decoding) is

$$\begin{aligned}\mathcal{R}_{\text{para}}(\mathbf{H}, \mathbf{I}) = \Bigg\{&(R_1,\ldots,R_L) \in \mathbb{R}^L_+ \ : \\ &R_\ell \leq \frac{1}{2}\log\left(1 + P_\ell\mathbf{h}_\ell^{\mathsf{T}}\Big(\mathbf{I} + \sum_{i\neq\ell} P_i\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}\Big)^{-1}\mathbf{h}_\ell\right)\Bigg\} \ ,\end{aligned}$$

which matches the rates attainable via i.i.d. Gaussian codebooks and treating interference as noise (see, e.g., [74, Equation 8.69]). As shown in [3] (for equal powers), the rate region for recovering all of the messages can be significantly expanded by *integer-forcing decoding*, i.e., taking the union over all rank-L matrices $\tilde{\mathbf{A}}$ in Theorem 1. See Section III-C for more details. ♢

As hinted in the example above, a receiver can recover all L transmitted messages if the coefficient matrix $\mathbf{Q} = [\mathbf{A}] \bmod p$ has rank L over $\mathbb{Z}_p$ and, under mild technical conditions, we can simply check if the integer coefficient matrix $\mathbf{A}$ itself has rank L over $\mathbb{R}$. More generally, to recover the mth message $\mathbf{w}_m$, the coefficient matrix must satisfy $\boldsymbol{\delta}_m^{\mathsf{T}} \in \text{rowspan}(\mathbf{Q})$ over $\mathbb{Z}_p$. It often more convenient to check whether $\delta_m^{\mathsf{T}} \in \text{rowspan}(\mathbf{A})$ over $\mathbb{R}$ and the following lemma gives a sufficient condition on when this is allowable.

Lemma 4: Consider the set of $L \times L$ integer matrices whose entries' magnitudes are upper bounded by a constant a_{max}. Let $\boldsymbol{\delta}_m^{\mathsf{T}}$ denote the mth row of the $L \times L$ identity matrix. If $\boldsymbol{\delta}_m^{\mathsf{T}} \in \text{rowspan}(\mathbf{A})$, then any rate tuple $(R_1,\ldots,R_L)$ in the computation rate region for $\mathbf{A}$ from Theorem 1 (or Theorem 2 below) is also achievable for recovering message $\mathbf{w}_m$. □

Roughly speaking, for large enough[10] p, it can be shown that $\boldsymbol{\delta}_m^{\mathsf{T}} \in \text{rowspan}(\mathbf{A})$ over the reals implies $\boldsymbol{\delta}_m^{\mathsf{T}} \in \text{rowspan}\big([\mathbf{A}] \bmod p\big)$ over $\mathbb{Z}_p$ since the entries are integer-valued and bounded. The proof follows along the same lines as that of [9, Theorem 5] and is omitted due to space considerations.

Remark 6: From Lemma 3, it suffices to check integer matrices $\mathbf{A}$ whose entries' magnitudes are upper bounded by $\lambda_{\text{max}}\big(\mathbf{I} + \mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)$ for Theorem 1. Since we have imposed an upper bound on the maximum singular value of the channel

[9] Technically speaking, our expression of the achievable rate region slightly generalizes the region described in [1] since we explicitly take a union over the set of integer matrices $\tilde{\mathbf{A}}$ that contain the rowspan of $\mathbf{A}$. This possibility is discussed in [1, Remark 7] but not formally included in the statement of [1, Theorem 1].

[10] Recall that the field size p can be chosen as large as desired according to Definition 4.

matrix $\mathbf{H}$ in Section II-B, we automatically obtain an upper bound on the entries of all viable $\mathbf{A}$. Furthermore, for scenarios where some probability of outage is permitted, it can be argued that it suffices for the probability density function of the largest singular value to have a vanishing tail. See [1, Remark 10] for more details. ◊

In certain scenarios, it may be useful to recover the integer-linear combination of the codewords (rather than the linear combination of the messages). For instance, two relays in a network may wish to simultaneously transmit a linear function of the codewords to benefit from a coherent gain [75]. Additionally, in a single-hop network, it is often more convenient to work directly with the codewords and ignore the finite field perspective.

Lemma 5: Under the nested lattice coding framework employed for Theorem 1 (and Theorem 2 in the next subsection), if the linear combinations $\mathbf{u}_1, \dots, \mathbf{u}_L$ with integer coefficient matrix $\mathbf{A}$ can be successfully recovered, then the integer-linear combinations of the codewords $\mathbf{a}_1^\mathsf{T}\mathbf{X}, \dots, \mathbf{a}_L^\mathsf{T}\mathbf{X}$ can be successfully recovered as well. □

This follows directly from Lemma 11 in Section VI. As a quick example, consider the scenario from Example 1. If the receiver can recover the modulo sum of the messages $\bigoplus_\ell \tilde{\mathbf{w}}_\ell$, then it can also recover the real sum of the codewords $\sum_\ell \mathbf{x}_\ell$.

Remark 7: Notice that, for a single-hop network, Lemma 5 allows us to directly check recoverability conditions over $\mathbb{R}$ instead of $\mathbb{Z}_p$. For instance, consider an integer matrix $\mathbf{A}$ such that (i) $\boldsymbol{\delta}_m^\mathsf{T} \in \operatorname{rowspan}(\mathbf{A})$ over $\mathbb{R}$ and (ii) $\boldsymbol{\delta}_m^\mathsf{T} \notin \operatorname{rowspan}\big([\mathbf{A}] \bmod p\big)$ over $\mathbb{Z}_p$. The latter condition means that it is not possible to retrieve the mth message from the recovered linear combinations. However, using Lemma 5, we can first recover the integer-linear combinations of the codewords $\mathbf{AX}$, solve for the mth codeword $\mathbf{x}_m$ over $\mathbb{R}$, and then infer the mth message $\mathbf{w}_m$ from $\mathbf{x}_m$. (This is not always possible in a multi-hop network since the destination may not have access to the channel observations that were used to recover the linear combinations.) ◊

B. Successive Computation

Consider a classical receiver that recovers individual codewords in a certain order. It is well-known that, once a codeword has been successfully decoded, it is beneficial to remove it from the channel observation so that subsequent decoding steps encounter less interference. Here, we explore an analogue of this successive interference cancellation (SIC) technique for compute-and-forward.

Assume that the receiver has (correctly) decoded the linear combinations with integer coefficient vectors $\mathbf{a}_1, \dots, \mathbf{a}_{m-1}$. These linear combinations can be used as side information at the receiver for two different forms of successive cancellation. The first reduces the effective noise variance and the second reduces the number of users that need to tolerate this effective noise.

Remark 8: Without loss of generality, we restrict ourselves to the decoding order $1, 2, \dots, L$. Any other decoding order can be mimicked by permuting the rows of the integer coefficient matrix $\mathbf{A}$. ◊

We begin by showing how the side information can be employed to decrease the effective noise variance. From Lemma 5, we know that the receiver has access to the integer-linear combinations of the codewords $\mathbf{a}_1^\mathsf{T}\mathbf{X}, \dots, \mathbf{a}_{m-1}^\mathsf{T}\mathbf{X}$, which we will write concisely as $\mathbf{A}_{m-1}\mathbf{X}$ where

$$\mathbf{A}_{m-1} \triangleq \begin{bmatrix} \mathbf{a}_1^\mathsf{T} \\ \vdots \\ \mathbf{a}_{m-1}^\mathsf{T} \end{bmatrix} .$$

Operationally, the receiver forms the effective channel

$$\begin{aligned} \tilde{\mathbf{y}}_m &= \mathbf{b}_m^\mathsf{T}\mathbf{Y} + \mathbf{c}_m^\mathsf{T}\mathbf{A}_{m-1}\mathbf{X} \\ &= \mathbf{a}_m^\mathsf{T}\mathbf{X} \ + \ \underbrace{(\mathbf{b}_m^\mathsf{T}\mathbf{H} + \mathbf{c}_m^\mathsf{T}\mathbf{A}_{m-1} - \mathbf{a}_m^\mathsf{T})\mathbf{X} + \mathbf{b}_m^\mathsf{T}\mathbf{Z}}_{\text{effective noise}} , \end{aligned}$$

where $\mathbf{b}_m \in \mathbb{R}^{N_\mathrm{r}}$ and $\mathbf{c}_m \in \mathbb{R}^{m-1}$ are equalization vectors. We denote the effective noise variance for successive computation by

$$\begin{aligned} &\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1}) \\ &\triangleq \frac{1}{n}\mathbb{E}\big\|(\mathbf{b}_m^\mathsf{T}\mathbf{H} + \mathbf{c}_m^\mathsf{T}\mathbf{A}_{m-1} - \mathbf{a}_m^\mathsf{T})\mathbf{X} + \mathbf{b}_m^\mathsf{T}\mathbf{Z}\big\|^2 \\ &= \|\mathbf{b}_m\|^2 + \big\|(\mathbf{b}_m^\mathsf{T}\mathbf{H} + \mathbf{c}_m^\mathsf{T}\mathbf{A}_{m-1} - \mathbf{a}_m^\mathsf{T})\mathbf{P}^{1/2}\big\|^2 . \end{aligned} \tag{12}$$

Note that, so long as $\mathbf{a}_m^\mathsf{T}$ is not orthogonal to $\operatorname{rowspan}(\mathbf{A}_{m-1})$, we can select $\mathbf{c}_m$ to obtain a strictly lower effective noise variance than possible via parallel computation (9).

Lemma 6: Assume that $\operatorname{rank}(\mathbf{A}_{m-1}) = m-1$. (Otherwise, delete rows of $\mathbf{A}_{m-1}$ until it has full rank and ignore the associated linear combinations in the decoding scheme.) The equalization vectors $\mathbf{b}_m \in \mathbb{R}^{N_\mathrm{r}}$ and $\mathbf{c}_m \in \mathbb{R}^{m-1}$ that minimize the effective noise variance from (12) are the MMSE projection vectors

$$\begin{aligned} \mathbf{b}_{\text{opt},m}^\mathsf{T} &= (\mathbf{a}_m^\mathsf{T} - \mathbf{c}_{\text{opt},m}^\mathsf{T}\mathbf{A}_{m-1})\mathbf{PH}^\mathsf{T}\big(\mathbf{I} + \mathbf{HPH}^\mathsf{T}\big)^{-1} \\ \mathbf{c}_{\text{opt},m}^\mathsf{T} &= \mathbf{a}_m^\mathsf{T}\mathbf{F}^\mathsf{T}\mathbf{F}\mathbf{A}_{m-1}^\mathsf{T}\big(\mathbf{A}_{m-1}\mathbf{F}^\mathsf{T}\mathbf{F}\mathbf{A}_{m-1}^\mathsf{T}\big)^{-1} , \end{aligned}$$

where $\mathbf{F}$ is a matrix that satisfies (7). Let

$$\mathbf{N}_{m-1} \triangleq \mathbf{I} - \mathbf{F}\mathbf{A}_{m-1}^\mathsf{T}\big(\mathbf{A}_{m-1}\mathbf{F}^\mathsf{T}\mathbf{F}\mathbf{A}_{m-1}^\mathsf{T}\big)^{-1}\mathbf{A}_{m-1}\mathbf{F}^\mathsf{T} \tag{13}$$

denote the projection matrix for the nullspace of $\mathbf{F}\mathbf{A}_{m-1}^\mathsf{T}$. The minimal effective noise variance is

$$\begin{aligned} &\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m | \mathbf{A}_{m-1}) \\ &\triangleq \sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_{\text{opt},m}, \mathbf{c}_{\text{opt},m} | \mathbf{A}_{m-1}) \\ &= \mathbf{a}_m^\mathsf{T}\mathbf{F}^\mathsf{T}\mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m \\ &= \big\|\mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m\big\|^2 . \end{aligned} \tag{14}$$

□

The proof can be found in Appendix C.

The second form of successive cancellation utilizes the algebraic structure of the codebooks, and was originally proposed in [9]. Recall that, in the parallel computation scheme, every fine lattice that participates in $\mathbf{a}_m^\mathsf{T}\mathbf{X}$ must be able to tolerate the associated effective noise. Clearly, if we knew some of the individual codewords, we could remove them from the channel observation and thus relax the noise tolerance constraints on their fine lattices. However, we only have access to certain linear combinations of the codewords, but it turns out that this is still enough side information to relax the noise tolerance constraints in a similar fashion.

In Section VI, we provide a detailed description of this *algebraic successive cancellation* technique. At a high level, it can be viewed as performing Gaussian elimination over $\mathbb{Z}_p$ where row swaps are not permitted. The following definition will be used to specify valid user cancellation orders.

Definition 5 (Admissible Mappings): Let $\mathbf{A}$ denote an $L \times L$ integer matrix and let $\mathcal{I} \subset \{1, \dots, L\} \times \{1, \dots, L\}$ denote a set of index pairs. We say that $\mathcal{I}$ is an *admissible mapping* for $\mathbf{A}$ if there exists a real-valued, lower unitriangular[11] matrix $\mathbf{L} \in \mathbb{R}^{L \times L}$ such that the (m, ℓ)th entry of $\mathbf{LA}$ is equal to zero for all $(m, \ell) \notin \mathcal{I}$. Let $\mathcal{M}(\mathbf{A})$ denote the set of admissible mappings for $\mathbf{A}$. ♢

Within the context of our scheme, if $\mathcal{I}$ is an admissible mapping for $\mathbf{A}$ and $(m, \ell) \in \mathcal{I}$, then user ℓ will not be cancelled out during the mth decoding step. Therefore, the ℓth transmitter must be able to tolerate the mth effective noise. The following theorem makes this notion precise.

Theorem 2: For an AWGN network with L transmitters, a receiver, and power constraints $P_1, \dots, P_L$, the following computation rate region is achievable,

$$\mathcal{R}_{\text{comp}}^{(\text{succ})}(\mathbf{H}, \mathbf{A}) = \bigcup_{\substack{\tilde{\mathbf{A}} \in \mathbb{Z}^{L \times L} \\ \text{rowspan}(\mathbf{A}) \subseteq \text{rowspan}(\tilde{\mathbf{A}})}} \ \bigcup_{\mathcal{I} \in \mathcal{M}(\tilde{\mathbf{A}})} \mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I})$$

where

$$\mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I}) \triangleq \Bigg\{ (R_1, \dots, R_L) \in \mathbb{R}_+^L \ : \qquad (15)$$

$$R_\ell \leq \frac{1}{2} \log^+ \left(\frac{P_\ell}{\sigma_{\text{succ}}^2(\mathbf{H}, \tilde{\mathbf{a}}_m | \tilde{\mathbf{A}}_{m-1})} \right) \ \forall (m, \ell) \in \mathcal{I} \Bigg\}$$

and $\tilde{\mathbf{a}}_m^{\mathsf{T}}$ and $\tilde{a}_{m,\ell}$ are the mth row and (m, ℓ)th entry of $\tilde{\mathbf{A}}$, respectively. □

The proof is presented in Section VI.

Corollary 1: For every channel matrix $\mathbf{H}$ and integer coefficient matrix $\mathbf{A}$, the parallel computation rate region is contained within the successive computation rate region, $\mathcal{R}_{\text{comp}}^{(\text{para})}(\mathbf{H}, \mathbf{A}) \subseteq \mathcal{R}_{\text{comp}}^{(\text{succ})}(\mathbf{H}, \mathbf{A})$. Furthermore, there exist $\mathbf{H}$ and $\mathbf{A}$ for which the subset relation is strict. □

Example 3: Consider three transmitters with equal power constraints, $P_1 = P_2 = P_3 = P$, and a receiver that observes $Y = 2X_1 + X_2 + X_3 + Z$ and wants to decode the linear combinations with integer coefficient vectors $\mathbf{a}_1 = [1 \ 1 \ 1]^{\mathsf{T}}$ and $\mathbf{a}_2 = [1 \ -1 \ -1]^{\mathsf{T}}$. In our notation, this corresponds to channel matrix $\mathbf{H} = [2 \ 1 \ 1]$ and integer matrix $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{0}]^{\mathsf{T}}$. The effective noise variances for successively decoding these linear combinations are

$$\sigma_{\text{succ}}^2(\mathbf{H}, \mathbf{a}_1) = \frac{3P + 2P^2}{1 + 6P}$$
$$\sigma_{\text{succ}}^2(\mathbf{H}, \mathbf{a}_2 | \mathbf{a}_1) = \frac{P(3 + 24P + 18P^2)}{8 + 38P + 36P^2} \ .$$

Following Definition 5, the mappings

$$\mathcal{I}_1 = \big\{(1,1),\ (1,2),\ (1,3),\ (2,2),\ (2,3)\big\}$$
$$\mathcal{I}_2 = \big\{(1,1),\ (1,2),\ (1,3),\ (2,1)\big\}$$

are admissible using the lower unitriangular matrices

$$\mathbf{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \ ,$$

respectively. This yields the rate regions

$$\mathcal{R}_{\text{succ}}(\mathbf{H}, \mathbf{A}, \mathcal{I}_1) = \Bigg\{ (R_1, R_2, R_3) \in \mathbb{R}_+^3 : \max_{\ell = 1,2,3} R_\ell \leq \frac{1}{2} \log^+ \left(\frac{1 + 6P}{3 + 2P} \right)$$
$$R_1 \leq \frac{1}{2} \log^+ \left(\frac{8 + 38P + 36P^2}{3 + 24P + 18P^2} \right) \Bigg\}$$

$$\mathcal{R}_{\text{succ}}(\mathbf{H}, \mathbf{A}, \mathcal{I}_2) = \Bigg\{ (R_1, R_2, R_3) \in \mathbb{R}_+^3 : \max_{\ell = 1,2,3} R_\ell \leq \frac{1}{2} \log^+ \left(\frac{1 + 6P}{3 + 2P} \right),$$
$$\max_{\ell = 2,3} R_\ell \leq \frac{1}{2} \log^+ \left(\frac{8 + 38P + 36P^2}{3 + 24P + 18P^2} \right) \Bigg\} \ ,$$

which are part of the achievable computation rate region expressed in Theorem 2. Note that the linear combination with integer coefficient vector $\mathbf{a}_2$ cannot be directly decoded using Theorem 1 since $\sigma_{\text{para}}^2(\mathbf{H}, \mathbf{a}_2) = 3P > P$. ♢

Example 4: We return to the scenario of Example 2 wherein the receiver wants all of the messages, $\mathbf{A} = \mathbf{I}$. Clearly, the mapping $\mathcal{I} = \big\{(1,1), \dots, (L,L)\big\}$ is admissible using the lower unitriangular matrix $\mathbf{L} = \mathbf{I}$. Consider the mth decoding step. The receiver has successfully decoded the first $m-1$ messages corresponding to $\mathbf{A}_{m-1} = [\boldsymbol{\delta}_1 \ \cdots \ \boldsymbol{\delta}_{m-1}]^{\mathsf{T}}$ where $\boldsymbol{\delta}_i$ is the ith column of $\mathbf{I}$. By setting the ith entry of the equalization vector $\mathbf{c}_m$ to be $\mathbf{b}_m^{\mathsf{T}} \mathbf{h}_i$ (and the rest to 0), the effective noise variance from (12) will be reduced to

$$\|\mathbf{b}_m\|^2 + \big\| \big(\mathbf{b}_m^{\mathsf{T}} \big[\mathbf{0} \ \cdots \ \mathbf{0} \ \mathbf{h}_m \ \cdots \ \mathbf{h}_L \big] - \boldsymbol{\delta}_m^{\mathsf{T}} \big) \mathbf{P}^{1/2} \big\|^2 \ .$$

Following the same steps as in Example 2, it can be shown that we can reach an effective SNR of

$$1 + P_\ell \mathbf{h}_\ell^{\mathsf{T}} \Bigg(\mathbf{I} + \sum_{i = \ell + 1}^{L} P_i \mathbf{h}_i \mathbf{h}_i^{\mathsf{T}} \Bigg)^{-1} \mathbf{h}_\ell$$

for the ℓth user. Thus, the rate region (for successive decoding) is

$$\mathcal{R}_{\text{succ}}(\mathbf{H}, \mathbf{I}) = \Bigg\{ (R_1, \dots, R_L) \in \mathbb{R}_+^L \ :$$
$$R_\ell \leq \frac{1}{2} \log \left(1 + P_\ell \mathbf{h}_\ell^{\mathsf{T}} \Bigg(\mathbf{I} + \sum_{i = \ell + 1}^{L} P_i \mathbf{h}_i \mathbf{h}_i^{\mathsf{T}} \Bigg)^{-1} \mathbf{h}_\ell \right) \Bigg\} \ .$$

This is equal to the rate region attainable via i.i.d. Gaussian codebooks and SIC decoding [76, Theorem 1] under the lexicographic decoding order. The SIC rate region for any decoding order can be attained via Theorem 2 by setting $\tilde{\mathbf{A}}$ to be the corresponding permutation matrix.

[11] A *unitriangular* matrix is a triangular matrix whose diagonal entries are equal to 1.

As shown in [7] (for equal powers), the rate region can be enlarged via *successive integer-forcing decoding*, i.e., taking the union over all full-rank matrices $\tilde{\mathbf{A}}$ in Theorem 2. Specifically, the successive integer-forcing rate region strictly contains the union of the SIC rate regions across all decoding orders. (Of course, when time-sharing across decoding orders is permitted then SIC can attain the entire capacity region.) $\Diamond$

The statement of the computation rate region in Theorem 2 takes a union over all integer matrices whose rowspan contains that of $\mathbf{A}$. We now argue that it suffices to take this union over a certain subset of these matrices. As a motivating example, consider a receiver that wishes to recover a single linear combination with an integer coefficient vector $\mathbf{a} = [a_1 \ \dots \ a_L]^\mathsf{T}$ whose entries have a greatest common divisor larger than one, $\gcd(a_1, \dots, a_L) = \alpha > 1$. If the receiver attempts to decode the corresponding integer-linear combination directly, it will encounter an effective noise variance of $\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}) = \|\mathbf{Fa}\|^2$ according to (11). However, the receiver can instead decode the linear combination with integer coefficient vector $\tilde{\mathbf{a}} = \alpha^{-1}\mathbf{a}$, which will yield a lower effective noise variance of $\sigma^2_{\text{para}}(\mathbf{H}, \tilde{\mathbf{a}}) = \|\mathbf{F}\tilde{\mathbf{a}}\|^2 = \alpha^{-2}\|\mathbf{Fa}\|^2$. Afterwards, it can scale by α to recover its desired linear combination.

We now show to generalize this concept to decoding more than one linear combination. We will need the following definitions.

Definition 6 (Unimodular Matrix): A square integer matrix is *unimodular* if its determinant equal to $+1$ or -1. $\Diamond$

It can be shown that the inverse of a unimodular matrix is itself unimodular.

Definition 7 (Primitive Basis Matrix): An integer matrix $\mathbf{A} \in \mathbb{Z}^{L \times L}$ whose rank is equal to $M \leq L$ is said to be a *primitive basis matrix*[12] if it is of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_M \\ \mathbf{0}_{(L-M)\times L} \end{bmatrix}$$

where $\mathbf{A}_M$ is a full-rank, $M \times L$ integer matrix, and there exists an integer matrix $\mathbf{B} \in \mathbb{Z}^{(L-M)\times L}$ such that the matrix

$$\begin{bmatrix} \mathbf{A}_M \\ \mathbf{B} \end{bmatrix}$$

is unimodular. In other words, $\mathbf{A}$ can be completed to a unimodular matrix. $\Diamond$

The following theorem establishes that the computation rate region from Theorem 2 is unchanged if we only take the union over primitive basis matrices.

Theorem 3: Let

$$\mathcal{R}^{(\text{prim})}_{\text{comp}}(\mathbf{H}, \mathbf{A}) = \bigcup_{\substack{\tilde{\mathbf{A}} \in \mathbb{Z}^{L\times L} \\ \tilde{\mathbf{A}} \text{ primitive basis} \\ \text{rowspan}(\mathbf{A}) \subseteq \text{rowspan}(\tilde{\mathbf{A}})}} \ \bigcup_{\mathcal{I} \in \mathcal{M}(\tilde{\mathbf{A}})} \mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I})$$

where $\mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I})$ is defined in (15). This region is equal to the computation rate region from Theorem 2,

$$\mathcal{R}^{(\text{prim})}_{\text{comp}}(\mathbf{H}, \mathbf{A}) = \mathcal{R}^{(\text{succ})}_{\text{comp}}(\mathbf{H}, \mathbf{A}) \ .$$

$\square$

The proof is deferred to Appendix D.

Corollary 2: If the integer coefficient matrix $\mathbf{A} \in \mathbb{Z}^{L\times L}$ has rank L, it suffices to take the union in Theorem 2 over the set of all unimodular matrices, rather than the set of rank L integer matrices. $\square$

C. Computation for Multiple-Access

At a first glance, it may seem that the lattice-based compute-and-forward framework developed above is a poor fit for multiple-access communication. Specifically, consider a receiver that observes the sum of the transmitted signals. From Example 1, the receiver can decode the sum of the codewords at a high rate, but this alone is insufficient to discern the individual messages. On the other hand, if the transmitters employ (independent) random i.i.d. codebooks, then all message tuples will be mapped to different sums with high probability.

However, within the compute-and-forward framework, a receiver is not restricted to decoding a single linear combination. In fact, a natural multiple-access strategy is to decode L linearly independent linear combinations and then solve for the underlying messages. This approach was first proposed in [9] for the parallel computation strategy with equal powers, and it was demonstrated that, when combined with algebraic successive cancellation, it always operates within a constant gap of the multiple-access sum capacity. Subsequent work [7] showed that, under certain technical conditions, the successive computation strategy with equal powers can operate at the exact multiple-access sum capacity. Below, we extend these results to the unequal power setting.

Recall that the multiple-access capacity region for our channel model is

$$\mathcal{R}_{\text{MAC}}(\mathbf{H}) = \Bigg\{ (R_1, \dots, R_L) \in \mathbb{R}^L_+ : \qquad (16)$$
$$\sum_{i\in\mathcal{S}} R_i \leq \frac{1}{2}\log\det\big(\mathbf{I} + \mathbf{H}_\mathcal{S}\mathbf{P}_\mathcal{S}\mathbf{H}_\mathcal{S}^\mathsf{T}\big) \ \forall \mathcal{S} \subseteq \{1, \dots, L\} \Bigg\}$$

where $\mathbf{H}_\mathcal{S}$ refers to the submatrix consisting of the columns of $\mathbf{H}$ with indices in $\mathcal{S}$ and $\mathbf{P}_\mathcal{S}$ to the submatrix consisting of the entries of $\mathbf{P}$ whose row and column indices are in $\mathcal{S}$. The multiple-access sum capacity is simply $\frac{1}{2}\log\det\big(\mathbf{I}+\mathbf{HPH}^\mathsf{T}\big)$. Any rate tuple in the capacity region is achievable using i.i.d. Gaussian codebooks at the transmitters combined with joint typicality decoding at the receiver [78, Section 15.3.1].

In order to operate near the multiple-access sum capacity, we will need to uniquely map users to effective noise variances via the following definition.

Definition 8 (Admissible Multiple-Access Mappings): Let $\mathcal{I} \subset \{1, \dots, L\} \times \{1, \dots, L\}$ be an admissible mapping for integer matrix $\mathbf{A} \in \mathbb{Z}^{L\times L}$ according to Definition 5 and let $\mathbf{L} \in \mathbb{R}^{L\times L}$ be the associated lower unitriangular matrix. We say that $\mathcal{I}$ *allows a multiple-access permutation* $\pi : \{1, 2, \dots, L\} \to \{1, 2, \dots, L\}$ if $\mathbf{LA}$ is upper triangular after column permutation by π. $\Diamond$

Intuitively, if $\mathcal{I}$ allows a multiple-access permutation π, the scheme will remove the $(\pi^{-1}(m))$th codeword from its effective channel after the mth decoding step. This means

[12] Our choice of terminology is inspired by the fact that a matrix of this form can be viewed as the basis of a primitive sublattice (i.e., a sublattice that is formed by intersecting an L-dimensional lattice with a subspace of dimension $M < L$) and that any L-dimensional lattice basis is unimodular. See [77, Section 1.2] for more details.

that the $(\pi^{-1}(m))$th codeword only needs to tolerate the first m effective noise variances. The details of this scheme are presented in Section VI.

There is always at least one admissible multiple-access mapping for every full-rank matrix $\mathbf{A} \in \mathbb{Z}^{L\times L}$. For instance, one can apply LU factorization with column permutation only to find an appropriate $\mathbf{L}$, admissible mapping $\mathcal{I}$, and permutation π. Also, recall from Remark 7 that if the receiver can recover L linear combinations with a full-rank integer coefficient matrix $\mathbf{A} \in \mathbb{Z}^{L\times L}$, then it can solve for the original messages.

Parallel Computation for Multiple Access: First, we note that, in the parallel computation strategy from Theorem 1, each user must overcome the effective noise for all linear combinations in which it participates. A simple multiple-access strategy is to recover a rank-L set of linear combinations that are then solved for the original messages. This corresponds to setting $\mathbf{A} = \mathbf{I}$ in Theorem 1, leading to the following achievable rate region for multiple access:

$$\mathcal{R}_{\text{comp}}^{(\text{para})}(\mathbf{H},\mathbf{I}) = \Bigg\{ (R_1,\dots,R_L) \in \mathbb{R}_+^L \ : \ R_\ell \le \max_{\substack{\tilde{\mathbf{A}}\in\mathbb{Z}^{L\times L}\\ \operatorname{rank}(\tilde{\mathbf{A}})=L}} \ \min_{m: a_{m,\ell}\neq 0} \frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H},\tilde{\mathbf{a}}_m)}\right) \Bigg\} \ .$$

This can be viewed as a generalization of the integer-forcing achievable rate from [3, Theorem 3] to unequal powers. As shown in [3], this strategy has a good ensemble average (e.g., for Rayleigh fading), but there are specific choices of $\mathbf{H}$ for which the sum rate lies arbitrarily far from the sum capacity.

As argued in [9], by augmenting parallel computation with algebraic successive cancellation, we can operate within a constant gap of the sum capacity. Here, we extend this result to the unequal power setting. This corresponds to applying Theorem 2 and setting the equalization vectors for the side information to zero, $\mathbf{c}_m = \mathbf{0}$, so that (15) is replaced with

$$\mathcal{R}_{\text{ASC}}(\mathbf{H},\tilde{\mathbf{A}},\mathcal{I}) = \Bigg\{ (R_1,\dots,R_L) \in \mathbb{R}_+^L \ : \ R_\ell \le \frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H},\tilde{\mathbf{a}}_m)}\right) \text{ for all } (m,\ell)\in\mathcal{I} \Bigg\} \ .$$

The next definition and lemma are taken from [14, Section VIII] and will be used to select appropriate integer vectors for parallel computation.

Definition 9 (Dominant Solutions): Let $\mathbf{F}$ be any matrix satisfying (7). A set of linearly independent integer vectors $\mathbf{a}_1^*,\dots,\mathbf{a}_L^* \in \mathbb{Z}^L$ satisfying $\|\mathbf{F}\mathbf{a}_1^*\| \le \cdots \le \|\mathbf{F}\mathbf{a}_L^*\|$ is called a *dominant solution* if, for any linearly independent integer vectors $\tilde{\mathbf{a}}_1,\dots,\tilde{\mathbf{a}}_L \in \mathbb{Z}^L$ satisfying $\|\mathbf{F}\tilde{\mathbf{a}}_1\| \le \cdots \le \|\mathbf{F}\tilde{\mathbf{a}}_L\|$, we have that $\|\mathbf{F}\mathbf{a}_m^*\| \le \|\mathbf{F}\tilde{\mathbf{a}}_m\|$ for $m=1,\dots,L$. We will call an integer matrix $\mathbf{A}^* \in \mathbb{Z}^{L\times L}$ a dominant solution if its rows $(\mathbf{a}_1^*)^\mathsf{T},\dots,(\mathbf{a}_L^*)^\mathsf{T}$ correspond to a dominant solution. ◊

Lemma 7 ([14, Theorem 8]): For any $\mathbf{F}$ satisfying (7), there always exists a dominant solution $\mathbf{a}_1^*,\mathbf{a}_2^*,\dots,\mathbf{a}_L^* \in \mathbb{Z}^L$ satisfying

$$\mathbf{a}_1^* = \arg\min\left\{\|\mathbf{F}\mathbf{a}\| : \mathbf{a}\in\mathbb{Z}^L\setminus\{\mathbf{0}\}\right\}$$
$$\mathbf{a}_2^* = \arg\min\left\{\|\mathbf{F}\mathbf{a}\| : \mathbf{a}\in\mathbb{Z}^L,\ \mathbf{a},\mathbf{a}_1^* \text{ linearly independent}\right\}$$
$$\vdots$$
$$\mathbf{a}_L^* = \arg\min\left\{\|\mathbf{F}\mathbf{a}\| : \mathbf{a}\in\mathbb{Z}^L,\ \mathbf{a},\mathbf{a}_1^*,\dots,\mathbf{a}_{L-1}^* \text{ linearly independent}\right\} .$$

□

See [14] for a proof as well as a greedy algorithm.

We can now show that the parallel computation strategy, when combined with algebraic successive cancellation, is approximately optimal for multiple access.

Theorem 4: For any dominant solution $\mathbf{A}^* \in \mathbb{Z}^{L\times L}$ and admissible multiple-access mapping $\mathcal{I}$ with permutation π, the rate tuple

$$R_\ell = \frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H},\mathbf{a}^*_{\pi(\ell)})}\right) \quad \ell=1,\dots,L$$

is achievable via the parallel computation strategy combined with algebraic successive cancellation, $(R_1,\dots,R_L) \in \mathcal{R}_{\text{ASC}}(\mathbf{H},\mathbf{A}^*,\mathcal{I})$. The sum of these rates is within a constant gap of the multiple-access sum capacity,

$$\sum_{\ell=1}^{L} R_\ell \ge \frac{1}{2}\log\det\left(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T}\right) - \frac{L}{2}\log(L) \ .$$

□

The proof can be found in Appendix E.

Successive Computation for Multiple Access: It is well-known that the corner points of the Gaussian multiple-access capacity region can be attained using i.i.d. Gaussian codebooks at the transmitters along with SIC decoding [76, Theorem 1]. As we will argue below, successive computation enjoys a similar optimality property: there is always at least one integer matrix for which the successive computation rate region includes a rate tuple that attains the multiple-access sum capacity. For instance, as shown in Example 4, successive computation decoding can mimic SIC decoding and attain the corner points of the capacity region. Furthermore, the successive computation rate region often includes non-corner points that attain the sum capacity (i.e., rate tuples that lie on the interior of the dominant face of the multiple-access capacity region). These points are not directly accessible via SIC decoding (but can be attained by enhancing SIC with time-sharing [79, Section 4.4] or rate-splitting [80]).

Our optimality results stem from the following identity.

Lemma 8: For any unimodular matrix $\mathbf{A} \in \mathbb{Z}^{L\times L}$ and permutation π, we have that

$$\sum_{\ell=1}^{L}\frac{1}{2}\log\left(\frac{P_\ell}{\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_{\pi(\ell)}|\mathbf{A}_{\pi(\ell)-1})}\right) = \frac{1}{2}\log\det\left(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T}\right) \ . \tag{17}$$

□

See Appendix F for a proof.

At a first glance, it may appear that Lemma 8 implies that *every* unimodular matrix yields sum-rate optimal performance for successive computation. However, this lemma does not guarantee that the rates appearing in (17) are achievable.

Specifically from (15), for an admissible mapping $\mathcal{I}$ that allows multiple-access permutation π, decoding is possible if the ℓth user can tolerate effective noise of variance

$$\max_{m:(m,\ell)\in\mathcal{I}} \sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m|\mathbf{A}_{m-1}) \ .$$

In order to apply Lemma 8 and show that Theorem 2 attains the exact sum capacity, this maximum must be equal to $\sigma^2_{\text{succ}}\big(\mathbf{H},\mathbf{a}_{\pi(\ell)}|\mathbf{A}_{\pi(\ell)-1}\big)$ for $\ell = 1,\dots,L$. Moreover, the achievable rate expression in (15) are written in terms of the $\log^+$ function whereas the summands in (17) simply use the $\log$ function. Thus, each user's power should exceed its associated effective noise variance. Putting these two conditions together and applying Lemma 8, we arrive at the following optimality result for successive computation.

Theorem 5: Let $\mathbf{A}$ be an $L\times L$ unimodular matrix and let $\mathcal{I}$ be an admissible mapping that allows multiple-access permutation π. If, for $\ell=1,\dots,L$, we have that

$$\max_{m:(m,\ell)\in\mathcal{I}} \sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m|\mathbf{A}_{m-1}) = \sigma^2_{\text{succ}}\big(\mathbf{H},\mathbf{a}_{\pi(\ell)}\big|\mathbf{A}_{\pi(\ell)-1}\big) \tag{18}$$

and $P_\ell \geq \sigma^2_{\text{succ}}\big(\mathbf{H},\mathbf{a}_{\pi(\ell)}\big|\mathbf{A}_{\pi(\ell)-1}\big)$, then the rate tuple

$$R_\ell = \frac{1}{2}\log\left(\frac{P_\ell}{\sigma^2_{\text{succ}}\big(\mathbf{H},\mathbf{a}_{\pi(\ell)}\big|\mathbf{A}_{\pi(\ell)-1}\big)}\right) \qquad \ell=1,\dots,L$$

is achievable via the successive computation strategy, $(R_1,\dots,R_L)\in\mathcal{R}_{\text{succ}}(\mathbf{H},\mathbf{A},\mathcal{I})$. Moreover, the sum of these rates is equal to the multiple-access sum capacity,

$$\sum_{\ell=1}^{L} R_\ell = \frac{1}{2}\log\det\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big) \ .$$

□

Remark 9: It is sometimes convenient to replace the condition in (18) with the stricter condition that

$$\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_1) \leq \sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_2|\mathbf{A}_1) \leq \cdots \leq \sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_L|\mathbf{A}_{L-1}) \ .$$

◊

Remark 10: For any channel matrix $\mathbf{H}$ and power matrix $\mathbf{P}$, there always exists a unimodular matrix $\mathbf{A}$ and multiple-access mapping $\mathcal{I}$ with permutation π for which Theorem 5 applies. For instance, as shown in Example 4, we can attain the performance for SIC decoding with order π by settting $\mathbf{A}$ to be the associated permutation matrix and $\mathcal{I} = \{(1,\pi(1)),\dots,(L,\pi(L))\}$. The next example demonstrates that the successive computation strategy can attain sum-capacity rate tuples that do not correspond to SIC corner points. ◊

Two-User Example: Consider a two-user multiple-access channel (i.e., $L=2$) with channel matrix $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2]$. For $\mathbf{H} = [1\ \frac{3}{2}]$, $P_1 = 7$, and $P_2 = 4$, we have plotted, in Figure 4, the capacity region (16) and marked the rate tuples achievable via SIC decoding, parallel computation for multiple access from Theorem 4, and successive computation for multiple access from Theorem 5. Specifically, SIC decoding yields the rates pairs $(0.3828, 1.6610)$ and $(1.5000, 0.5437)$. Successive computation can attain both of these rate pairs as well as $(1.0850, 0.9588)$ and $(1.3624, 0.6813)$. Finally, there are two dominant rate pairs for parallel computation: $(1.3624, 0.5903)$ and $(0.9940, 0.9588)$. Note that all successive computation rate pairs are sum-capacity optimal and that the parallel computation rate pairs are much closer to the sum-capacity than the (worst-case) bound of 2 bits.[13] ◊

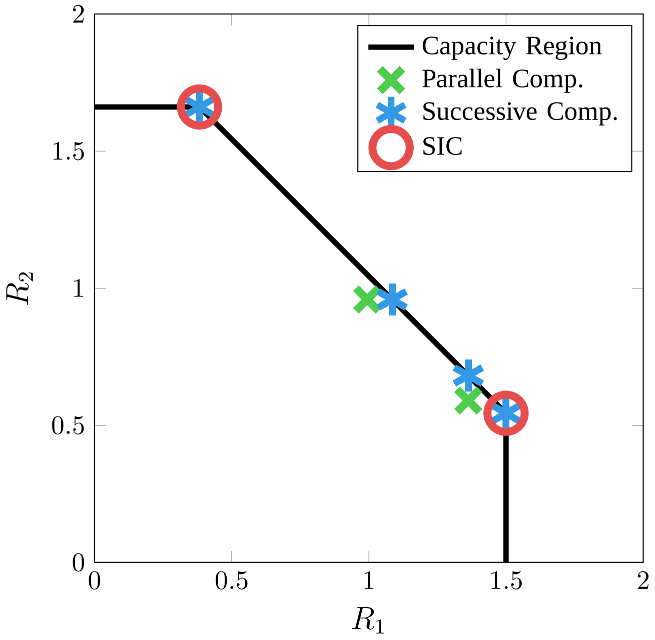

Fig. 4. Plot of the Gaussian multiple-access capacity region as well as the dominant rate pairs for parallel computation, sum-capacity optimal rate pairs for successive computation, and SIC corner points for the channel matrix $\mathbf{H} = [1\ \frac{3}{2}]$ and powers $P_1 = 7$ and $P_2 = 4$.

Remark 11: In independent and concurrent work, Zhu and Gastpar have developed a compute-and-forward approach to multiple-access [62]. Their main idea is to use the same fine lattice at each transmitter along with different coarse lattices. The second moments of the coarse lattices are set according to the desired rates, and each transmitter scales its lattice codeword to meet its power constraint. (Note that their approach does not establish a correspondence to a finite field.) Overall, they establish that a significant fraction of the multiple-access sum-capacity boundary is achievable via this approach. We note that the underlying compute-and-forward result for unequal powers [62, Theorem 1] is a special case of Theorem 1. ◊

D. Multiple Receivers

So far, we have limited our discussion and results to single-receiver scenarios. Although this has allowed us to introduce our main ideas in a compact fashion, the compute-and-forward framework is the most useful in scenarios where there are *multiple receivers* that observe interfering codewords. We now expand our problem statement to permit K receivers, potentially with different demands. (See Figure 5 for a block diagram.) As we will see, the parallel and successive compu-

[13] MATLAB code to generate this plot for any (two-user) choice of $\mathbf{H}$ and $\mathbf{P}$ is available on the first author's website.

tation rate regions can be expressed in terms as intersections of the corresponding rate regions for each receiver.

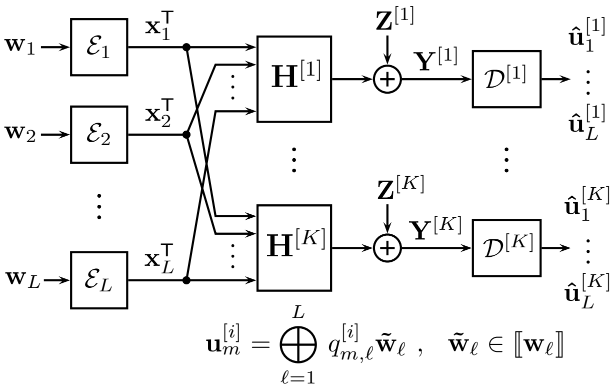
$\mathbf{w}_1$
$\mathcal{E}_1$
$\mathbf{x}_1^\mathsf{T}$
$\mathbf{w}_2$
$\mathcal{E}_2$
$\mathbf{x}_2^\mathsf{T}$
$\mathbf{w}_L$
$\mathcal{E}_L$
$\mathbf{x}_L^\mathsf{T}$
$\mathbf{H}^{[1]}$
$\mathbf{Z}^{[1]}$
$\mathbf{Y}^{[1]}$
$\mathcal{D}^{[1]}$
$\hat{\mathbf{u}}_1^{[1]}$
$\hat{\mathbf{u}}_L^{[1]}$
$\mathbf{H}^{[K]}$
$\mathbf{Z}^{[K]}$
$\mathbf{Y}^{[K]}$
$\mathcal{D}^{[K]}$
$\hat{\mathbf{u}}_1^{[K]}$
$\hat{\mathbf{u}}_L^{[K]}$
$\mathbf{u}_m^{[i]} = \bigoplus_{\ell=1}^{L} q_{m,\ell}^{[i]}\tilde{\mathbf{w}}_\ell \ , \quad \tilde{\mathbf{w}}_\ell \in [\![\mathbf{w}_\ell]\!]$

Fig. 5. Block diagram for the compute-and-forward problem with multiple receivers. Each transmitter has a message $\mathbf{w}_\ell$ whose elements are taken from $\mathbb{Z}_p$. This message is embedded into $\mathbb{Z}_p^k$ (by zero-padding), mapped into a codeword $\mathbf{x}_\ell \in \mathbb{R}^n$, and sent over the channel. Each receiver observes a noisy linear combination of these codewords, $\mathbf{Y}^{[i]} = \mathbf{H}^{[i]}[\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_L]^\mathsf{T} + \mathbf{Z}^{[i]}$, and attempts to recover the linear combinations $\mathbf{u}_1^{[i]}, \dots, \mathbf{u}_L^{[i]}$ of the coset representatives of the original messages.

As in Section II-B, for $i = 1, \dots, K$, the ith receiver is equipped with $N_\text{r}^{[i]}$ antennas and observes a noisy linear combination of the channel inputs,

$$\mathbf{Y}^{[i]} = \sum_{\ell=1}^{L} \mathbf{h}_\ell^{[i]} \mathbf{x}_\ell^\mathsf{T} + \mathbf{Z}^{[i]}$$

where $\mathbf{h}_\ell^{[i]} \in \mathbb{R}^{N_\text{r}^{[i]}}$ is the channel vector between the ℓth transmitter and the ith receiver and $\mathbf{Z}^{[i]}$ is elementwise i.i.d. $\mathcal{N}(0,1)$. We group the channel vectors corresponding to the ith receiver into a channel matrix

$$\mathbf{H}^{[i]} \triangleq \left[\mathbf{h}_1^{[i]} \quad \mathbf{h}_2^{[i]} \quad \cdots \quad \mathbf{h}_L^{[i]}\right] .$$

The problem statement is essentially the same as in Section II-D. Following Definition 3, we define a *decoder* $\mathcal{D}^{[i]} : \mathbb{R}^{N_\text{r}^{[i]} \times n} \times \mathbb{R}^{N_\text{r}^{[i]} \times L} \times \mathbb{Z}^{L \times L} \to \mathbb{Z}_p^{L \times k}$ for the ith receiver that takes as inputs the channel observation $\mathbf{Y}^{[i]}$, the channel matrix $\mathbf{H}^{[i]}$, and the desired integer coefficient matrix $\mathbf{A}^{[i]}$, and outputs an estimate $\hat{\mathbf{U}}^{[i]} = \mathcal{D}^{[i]}(\mathbf{Y}^{[i]}, \mathbf{H}^{[i]}, \mathbf{A}^{[i]})$. Let $(\hat{\mathbf{u}}_m^{[i]})^\mathsf{T}$ denote the mth row of $\mathbf{U}^{[i]}$. We say that decoding is *successful* if $\hat{\mathbf{u}}_1^{[i]} = \mathbf{u}_1^{[i]}, \dots, \hat{\mathbf{u}}_L^{[i]} = \mathbf{u}_L^{[i]}$ for some linear combinations of the form

$$\mathbf{u}_m^{[i]} = \bigoplus_{\ell=1}^{L} q_{m,\ell}^{[i]} \tilde{\mathbf{w}}_\ell$$

where the $q_{m,\ell}^{[i]} \in \mathbb{Z}_p$ are the entries of $\mathbf{Q}^{[i]} = [\mathbf{A}^{[i]}] \bmod p$ and $\tilde{\mathbf{w}}_\ell \in [\![\mathbf{w}_\ell]\!]$. An *error* occurs if decoding is not successful for some $i \in \{1, \dots, K\}$.

Remark 12: Note that we insist that the same coset representatives $\tilde{\mathbf{w}}_\ell$ are used across receivers. This is to ensure that the linear combinations from multiple receivers can later be put together to recover the original messages. $\Diamond$

We now adjust our definition of a computation rate region to accommodate multiple receivers.

Definition 10 (Multiple-Receiver Computation Rate Region): A *computation rate region for K receivers* is specified by a set-valued function $\mathcal{R}_\text{comp}\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}, \mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]}\big)$ that maps K-tuples of channel matrices $\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}\big) \in \mathbb{R}^{N_r^{[1]} \times L} \times \cdots \times \mathbb{R}^{N_r^{[K]} \times L}$ and K-tuples of integer coefficient matrices $\big(\mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]}\big) \in \mathbb{Z}^{L \times L} \times \cdots \times \mathbb{Z}^{L \times L}$ to a subset of $\mathbb{R}_+^L$. The computation rate region described by $\mathcal{R}_\text{comp}$ is *achievable* if, for every rate tuple $(R_1, R_2, \dots, R_L) \in \mathbb{R}_+^L$, $\epsilon > 0$, and n large enough, there exist

- parameters $p, k_{\text{C},\ell}, k_{\text{F},\ell}$ satisfying $\dfrac{k_{\text{F},\ell} - k_{\text{C},\ell}}{n} \log p > R_\ell - \epsilon$ for $\ell = 1, 2, \dots, L$,
- encoders $\mathcal{E}_1, \dots, \mathcal{E}_L$,

such that,

- for all K-tuples of channel matrices $\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}\big) \in \mathbb{R}^{N_r^{[1]} \times L} \times \cdots \times \mathbb{R}^{N_r^{[K]} \times L}$ and
- every K-tuple of integer matrices $(\mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]})$ satisfying $(R_1, \dots, R_L) \in \mathcal{R}_\text{comp}\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}, \mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]}\big)$

there exist decoders $\mathcal{D}_1, \dots, \mathcal{D}_K$ with probability of decoding error at most ϵ. $\Diamond$

We can now state the achievable computation rate regions for parallel computation and successive computation.

Theorem 6: For an AWGN network with L transmitters satisfying power constraints $P_1, P_2, \dots, P_L$, respectively, and K receivers, the following computation rate region is achievable,

$$\begin{aligned}&\mathcal{R}_\text{comp}^\text{(para)}\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}, \mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]}\big) \\ &= \bigcap_{i=1}^{K} \mathcal{R}_\text{comp}^\text{(para)}\big(\mathbf{H}^{[i]}, \mathbf{A}^{[i]}\big)\end{aligned}$$

for the function $\mathcal{R}_\text{comp}^\text{(para)}\big(\mathbf{H}^{[i]}, \mathbf{A}^{[i]}\big)$ as defined in Theorem 1. $\square$

The proof can be inferred from that of Theorem 1: the encoders implement the same scheme as in the single-receiver case and each receiver implements the parallel computation decoder described in Section V.

Theorem 7: For an AWGN network with L transmitters satisfying power constraints $P_1, P_2, \dots, P_L$, respectively, and K receivers, the following computation rate region is achievable,

$$\begin{aligned}&\mathcal{R}_\text{comp}^\text{(succ)}\big(\mathbf{H}^{[1]}, \dots, \mathbf{H}^{[K]}, \mathbf{A}^{[1]}, \dots, \mathbf{A}^{[K]}\big) \\ &= \bigcap_{i=1}^{K} \mathcal{R}_\text{comp}^\text{(succ)}\big(\mathbf{H}^{[i]}, \mathbf{A}^{[i]}\big)\end{aligned}$$

for $\mathcal{R}_\text{comp}^\text{(succ)}\big(\mathbf{H}^{[i]}, \mathbf{A}^{[i]}\big)$ as defined in Theorem 2. $\square$

Again, the proof can be inferred from that of Theorem 2: the encoders implement the same scheme as in the single-receiver case and each receiver implements the successive computation decoder described in Section VI.

Unfortunately, the multiple-access sum-capacity optimality results do not transfer directly from the single-receiver setting. Nonetheless, in the multiple-receiver setting, computation for multiple-access outperforms i.i.d. Gaussian encoding with SIC decoding even when time-sharing is allowed. We explore this phenomenon in the context of a compound multiple-access channel below.

E. Case Study: Two-User Gaussian Compound Multiple-Access Channel

We now take an in-depth look at the performance of SIC and compute-and-forward for a two-user Gaussian compound multiple-access channel. In our notation, this corresponds to $L = 2$ transmitters with messages $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively, and $K = 2$ receivers that both want to recover $\mathbf{w}_1$ and $\mathbf{w}_2$. The capacity region is the intersection of the individual MAC capacity regions,

$$\mathcal{R}_{\text{CMAC}}\big(\mathbf{H}^{[1]},\mathbf{H}^{[2]}\big) = \mathcal{R}_{\text{MAC}}\big(\mathbf{H}^{[1]}\big) \cap \mathcal{R}_{\text{MAC}}\big(\mathbf{H}^{[2]}\big)$$

and can be achieved via i.i.d. Gaussian coding and joint typicality decoding. As discussed below in Remark 14, the compound MAC often appears in the context of K-user interference channels. In some scenarios, the transmitters may opt to induce interference alignment by using lattice codebooks instead of i.i.d. Gaussian codebooks. This motivates the need for lattice-based decoding strategies.

In order for the ith receiver to successfully recover both messages with SIC decoding, the rates must fall within the SIC decoding region for the corresponding (two-user) multiple-access channel,

$$\begin{aligned}
&\mathcal{R}^{(\text{SIC})}_{\text{MAC}}\big(\mathbf{H}^{[i]}\big) = \mathcal{R}_{\text{SIC},a}\big(\mathbf{H}^{[i]}\big) \cup \mathcal{R}_{\text{SIC},b}\big(\mathbf{H}^{[i]}\big)\\
&\mathcal{R}_{\text{SIC},a}\big(\mathbf{H}^{[i]}\big) = \Big\{(R_1,R_2) \in \mathbb{R}^2_+ :\\
&\quad R_1 \leq \frac{1}{2}\log(1+\big\|\mathbf{h}^{[i]}_1\big\|^2 P_1),\\
&\quad R_2 \leq \frac{1}{2}\log\Big(1+P_2\big(\mathbf{h}^{[i]}_2\big)^{\mathsf{T}}\Big(\mathbf{I}+P_1\mathbf{h}^{[i]}_1\big(\mathbf{h}^{[i]}_1\big)^{\mathsf{T}}\Big)^{-1}\mathbf{h}^{[i]}_2\Big)\Big\}\\
&\mathcal{R}_{\text{SIC},b}\big(\mathbf{H}^{[i]}\big) = \Big\{(R_1,R_2) \in \mathbb{R}^2_+ :\\
&\quad R_1 \leq \frac{1}{2}\log\Big(1+P_1\big(\mathbf{h}^{[i]}_1\big)^{\mathsf{T}}\Big(\mathbf{I}+P_2\mathbf{h}^{[i]}_2\big(\mathbf{h}^{[i]}_2\big)^{\mathsf{T}}\Big)^{-1}\mathbf{h}^{[i]}_1\Big),\\
&\quad R_2 \leq \frac{1}{2}\log(1+\big\|\mathbf{h}^{[i]}_2\big\|^2 P_2)\Big\}\ .
\end{aligned}$$

Thus, for the compound multiple-access channel, SIC decoding combined with time-sharing can attain

$$\mathcal{R}^{(\text{SIC})}_{\text{CMAC}}\big(\mathbf{H}^{[1]},\mathbf{H}^{[2]}\big) = \operatorname{conv}\Big(\mathcal{R}^{(\text{SIC})}_{\text{MAC}}\big(\mathbf{H}^{[1]}\big) \cap \mathcal{R}^{(\text{SIC})}_{\text{MAC}}\big(\mathbf{H}^{[2]}\big)\Big)$$

where conv refers to the convex hull operation. Note that SIC decoding does not, in general, attain the sum-capacity even if aided by time-sharing. This is due to the fact that the corner points of the two multiple-access capacity regions do not coincide nor do the time-sharing ratios required to reach any other sum-capacity points.

Successive computation does not reach the sum capacity for similar reasons. Namely, the sum-capacity rate pairs that can be directly attained with successive computation (see Theorem 5) will differ across receivers as will the required time-sharing ratios for other sum-capacity points. Using the achievable computation rate region from Theorem 7 combined with time-sharing and setting $\mathbf{A}^{[1]} = \mathbf{A}^{[2]} = \mathbf{I}$, we get that

$$\begin{aligned}
&\mathcal{R}^{\text{succ}}_{\text{CMAC}}\big(\mathbf{H}^{[1]},\mathbf{H}^{[2]}\big)\\
&= \operatorname{conv}\Big(\mathcal{R}^{(\text{succ})}_{\text{comp}}\big(\mathbf{H}^{[1]},\mathbf{I}\big)\ \cap\ \mathcal{R}^{(\text{succ})}_{\text{comp}}\big(\mathbf{H}^{[2]},\mathbf{I}\big)\Big)
\end{aligned}$$

is achievable for the compound multiple-access channel. The flexibility to optimize over full-rank integer matrices $\tilde{\mathbf{A}}$ yields a larger rate region than SIC, as shown below.

Example 5: In Figure 6, we have illustrated how these rate regions are calculated for a compound multiple-access channel with $\mathbf{H}^{[1]} = [3.3\ 2.1]$, $\mathbf{H}^{[2]} = [2.4\ 4.2]$, $P_1 = 4$, and $P_2 = 3$. The corner points of the multiple-access capacity region with respect to receiver 1 are $(1.0109, 1.9154)$ and $(2.7388, 0.1875)$. Successive computation achieves these corner points as well as the rate pairs $(1.5084, 1.4180)$ and $(1.6255, 1.3008)$. With respect to receiver 2, the corner points are $(0.2566, 2.8764)$ and $(2.2937, 0.8393)$ and successive computation achieves these corner points as well as the rate pairs $(1.3799, 1.7531)$ and $(1.9606, 1.1724)$. As expected, after intersecting the individual multiple-access regions and time-sharing, neither SIC nor successive computation reach the sum capacity.

Remark 13: In recent work, Wang *et al.* [81] have demonstrated that, for this scenario, a variation of SIC that encodes messages across multiple blocks and employs sliding-window decoding can match the performance of joint typicality decoding. ♢

Remark 14: The compound multiple-access channel is an important building block for understanding the capacity region of multi-user interference networks. For instance, in the strong interference regime, the capacity region of a two-user interference channel corresponds exactly to that of a two-user compound multiple-access channel [82], [83]. In this two-user setting, successive computation is inferior to i.i.d. random coding with joint typicality decoding. However, in an interference channel with three or more users, there is the possibility of interference alignment [84], [85]. For instance, in the K-user symmetric Gaussian interference channel, the sum capacity in the strong interference regime can be approximated by that of a two-user symmetric Gaussian compound multiple-access channel [9]. To induce alignment at the receivers, the scheme from [9] employs the same lattice codebook at each transmitter and has each receiver decode its desired message indirectly, by first recovering two independent linear combinations via compute-and-forward. In recent work [10], we have shown that lattice interference alignment is possible in any setting where the beamforming vectors are aligned "stream-by-stream" so long as the codewords are allowed to have unequal powers.

IV. Nested Lattice Codes

In this section, we describe the nested lattice codes that will be the building blocks of our encoding and decoding schemes. We begin with some basic lattice definitions in Section IV-A, present nested lattice constructions and properties in Section IV-B, and discuss mappings to and from $\mathbb{Z}^k_p$ in Section IV-C.

A. Lattice Definitions

We now review some properties of lattices that will be useful for our code constructions and refer interested readers to the textbook of Zamir [60] for a comprehensive treatment of lattices for coding. A *lattice* Λ is a discrete additive subgroup of $\mathbb{R}^n$ that is closed under addition and reflection, i.e., for any

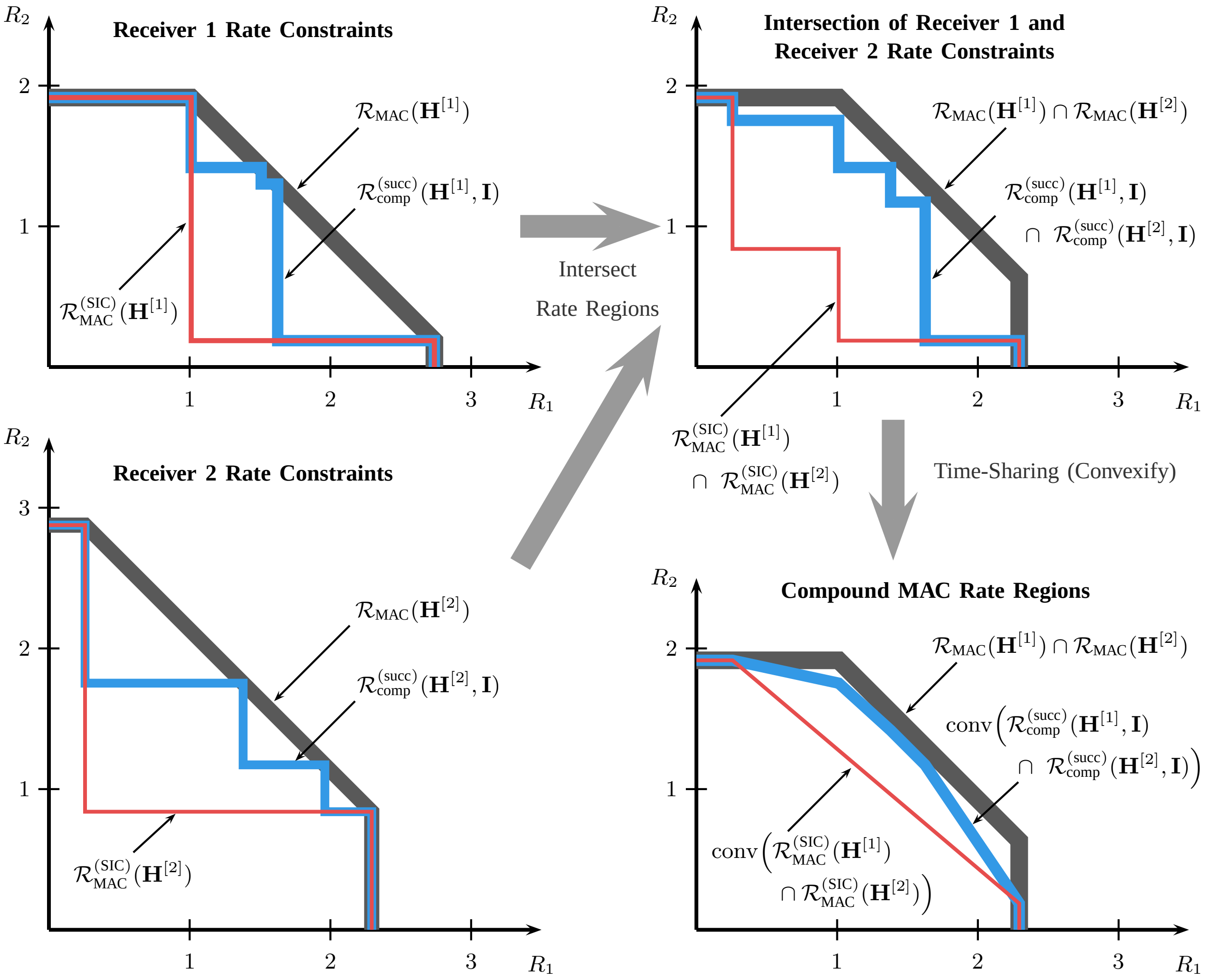

Fig. 6. Step-by-step illustration for determining the capacity region, successive computation rate region, and SIC rate region for a compound multiple-access channel with two transmitters with powers $P_1 = 4$ and $P_2 = 3$, respectively, and two receivers with channel matrices $\mathbf{H}^{[1]} = [3.3 \;\; 2.1]$ and $\mathbf{H}^{[2]} = [2.4 \;\; 4.2]$, respectively. The top left depicts the rate constraints that must be satisfied for the first receiver to successfully decode both messages and the bottom left depicts the corresponding rate constraints for the second receiver. The top right shows the intersection of these rate constraints and the bottom right shows the achievable rate regions for the compound multiple-access channel that comes from convexification of the intersected regions, which corresponds operationally to time-sharing. Note that no convexification is needed for the capacity region $\mathcal{R}_{\text{MAC}}(\mathbf{H}^{[1]}) \cap \mathcal{R}_{\text{MAC}}(\mathbf{H}^{[2]})$ since it is the intersection of two convex rate regions.

$\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda$, we have that $-\boldsymbol{\lambda}_1, -\boldsymbol{\lambda}_2 \in \Lambda$ and $\boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 \in \Lambda$. Note that this implies that the zero vector $\mathbf{0}$ is always an element of the lattice.

Let

$$Q_\Lambda(\mathbf{x}) \triangleq \arg\min_{\boldsymbol{\lambda}\in\Lambda} \|\mathbf{x} - \boldsymbol{\lambda}\|$$

denote the nearest neighbor quantizer for Λ. Using this, we define the (fundamental) *Voronoi region* $\mathcal{V}$ of Λ to be the set of all points in $\mathbb{R}^n$ which are quantized to the zero vector (breaking ties in a systematic fashion). We also define the *modulo operation*, which outputs the error from quantizing $\mathbf{x}$ onto Λ, as

$$[\mathbf{x}] \bmod \Lambda \triangleq \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$

The modulo operation satisfies a *distributive law*,

$$\big[a[\mathbf{x}] \bmod \Lambda + b[\mathbf{y}] \bmod \Lambda\big] \bmod \Lambda = [a\mathbf{x} + b\mathbf{y}] \bmod \Lambda \ ,$$

for any integers $a, b \in \mathbb{Z}$. The *second moment* of a lattice, denoted as $\sigma^2(\Lambda)$, is the second moment per dimension of the norm of a random vector that is drawn uniformly over the fundamental Voronoi region $\mathcal{V}$, that is,

$$\sigma^2(\Lambda) \triangleq \frac{1}{n}\int_{\mathcal{V}} \|\mathbf{x}\|^2 \frac{1}{\text{Vol}(\mathcal{V})} d\mathbf{x} \ ,$$

where $\text{Vol}(\mathcal{V})$ denotes the volume of $\mathcal{V}$.

The following lemma will be useful in characterizing the distributions of dithered lattice codewords.

Lemma 9 (Crypto Lemma): Let Λ be a lattice, $\mathbf{x}$ a random vector with an arbitrary distribution over $\mathbb{R}^n$, and $\mathbf{d}$ a random vector that is independent of $\mathbf{x}$ and uniform over $\mathcal{V}$. It follows that the random vector $[\mathbf{x}+\mathbf{d}] \bmod \Lambda$ is independent of $\mathbf{x}$ and uniform over $\mathcal{V}$. □

See [60, Ch. 4.1] for a detailed discussion and proof.

We say that lattice Λ_{C} is *nested* in lattice Λ_{F} if $\Lambda_{\text{C}} \subset \Lambda_{\text{F}}$. The lattice Λ_{F} is often referred to as the *fine* lattice and Λ_{C} as the *coarse* lattice. The coarse lattice induces a *partition* of the fine lattice into *cosets* of the form $\boldsymbol{\lambda} + \Lambda_{\text{C}}$ for $\boldsymbol{\lambda} \in \Lambda_{\text{F}}$. The set of all such cosets is written as $\Lambda_{\text{F}}/\Lambda_{\text{C}} \triangleq \{\boldsymbol{\lambda} + \Lambda_{\text{C}} : \boldsymbol{\lambda} \in \Lambda_{\text{F}}\}$.

(The notation $\Lambda_\text{F}/\Lambda_\text{C}$ refers to the fact that this is a quotient group.) It will often be useful to represent each coset by a single element, i.e., a *coset representative*. We will represent each coset by its minimum norm element to obtain a set of coset representatives,

$$[\Lambda_\text{F}/\Lambda_\text{C}] \triangleq \Lambda_\text{F} \bmod \Lambda_\text{C} \ .$$

A *nested lattice codebook* $\mathcal{L}$ is generated using a nested lattice pair $\Lambda_\text{C} \subset \Lambda_\text{F}$. The codebook $\mathcal{L}$ comprises all elements of the fine lattice that fall within the Voronoi region $\mathcal{V}_\text{C}$ of the coarse lattice, $\mathcal{L} = \Lambda_\text{F} \cap \mathcal{V}_\text{C}$. The rate of the codebook is

$$\frac{1}{n}\log|\mathcal{L}| = \frac{1}{n}\log\left(\frac{\text{Vol}(\mathcal{V}_\text{C})}{\text{Vol}(\mathcal{V}_\text{F})}\right)$$

where $\mathcal{V}_\text{F}$ is the Voronoi region of Λ_F. It can be shown that the set $\Lambda_\text{F} \cap \mathcal{V}_\text{C}$ is equal to the set of minimum-norm coset representatives $[\Lambda_\text{F}/\Lambda_\text{C}]$. This implies that the codebook $\mathcal{L}$ can be interpreted algebraically (i.e., as the set of coset representatives $[\Lambda_\text{F}/\Lambda_\text{C}]$) and geometrically (i.e., as the set of all elements of the fine lattice Λ_F that are in the Voronoi region of the coarse lattice Λ_F). Loosely speaking, the geometric properties of our lattice constructions will be useful in ensuring that the codewords are spaced sufficiently far apart in $\mathbb{R}^n$ and that the power constraints are satisfied (Theorem 8). Similarly, the algebraic interpretation will be useful for constructing a linear mapping between our nested lattice codebooks and $\mathbb{Z}_p^k$, the vector space from which the messages are drawn (Theorem 9).

Finally, note that any nested lattice pair $\Lambda_\text{C} \subset \Lambda_\text{F}$ satisfies the following quantization property:

$$\left[Q_{\Lambda_\text{F}}(\mathbf{x})\right] \bmod \Lambda_\text{C} = \Big[Q_{\Lambda_\text{F}}\big([\mathbf{x}] \bmod \Lambda_\text{C}\big)\Big] \bmod \Lambda_\text{C} \ . \quad (19)$$

B. Nested Lattice Constructions

We will employ the nested lattice construction of Ordentlich and Erez [59] as part of our achievability scheme. We will design nested lattice codebooks using the same parameters in our problem statement, $n, p, k_{\text{C},\ell}, k_{\text{F},\ell}$, for $\ell = 1, \ldots, L$. Consider the prime p and the corresponding finite field[14] $\mathbb{Z}_p$. The key idea is to take a series of nested linear codes of length n over $\mathbb{Z}_p$ and then lift the codes from $\mathbb{Z}_p^n$ to $\mathbb{R}^n$ using Construction A to obtain a series of nested lattices.

We now specify parameters[15] for which sequences of good nested lattices exist. Let $P_\text{max} \triangleq \max_\ell P_\ell$ and V_n be the volume of an n-dimensional ball of radius 1. Following the construction in [59], for a given blocklength n, powers P_ℓ, and noise tolerances $\sigma^2_{\text{eff},\ell}$, we will set p to be the largest prime between $\frac{1}{2}n^{3/2}$ and $n^{3/2}$, which is guaranteed to exist for $n > 1$ by Bertrand's Postulate[16], and

$$\gamma = 2\sqrt{nP_\text{max}2^\alpha}$$

$$k_{\text{C},\ell} = \frac{n}{2\log p}\left(\log\left(\frac{P_\text{max}}{P_\ell}\right) + \log\left(\frac{4}{V_n^{2/n}}\right) + \alpha\right) \quad (20)$$

$$k_{\text{F},\ell} = \frac{n}{2\log p}\left(\log\left(\frac{P_\text{max}}{\sigma^2_{\text{eff},\ell}}\right) + \log\left(\frac{4}{V_n^{2/n}}\right) + \alpha\right) \quad (21)$$

where $\alpha > 0$ will be chosen as part of Theorem 8.

Recall that $k_\text{F} \triangleq \max_\ell k_{\text{F},\ell}$ and consider a $k_\text{F} \times n$ matrix $\mathbf{G}$ over $\mathbb{Z}_p$. Let $\mathbf{G}_{\text{C},\ell}$ denote the matrix consisting of the first $k_{\text{C},\ell}$ rows of $\mathbf{G}$ and let $\mathbf{G}_{\text{F},\ell}$ denote the matrix consisting of the first $k_{\text{F},\ell}$ rows of $\mathbf{G}$ for $\ell = 1, \ldots, L$. Define $\mathcal{C}_{\text{C},\ell}$ and $\mathcal{C}_{\text{F},\ell}$ to be the vector spaces generated by taking the columns of $\mathbf{G}^\mathsf{T}_{\text{C},\ell}$ and $\mathbf{G}^\mathsf{T}_{\text{F},\ell}$ as a basis, respectively:

$$\mathcal{C}_{\text{C},\ell} = \left\{\mathbf{G}^\mathsf{T}_{\text{C},\ell}\mathbf{w} : \mathbf{w} \in \mathbb{Z}_p^{k_{\text{C},\ell}}\right\}$$
$$\mathcal{C}_{\text{F},\ell} = \left\{\mathbf{G}^\mathsf{T}_{\text{F},\ell}\mathbf{w} : \mathbf{w} \in \mathbb{Z}_p^{k_{\text{F},\ell}}\right\} \ .$$

Note that $\mathcal{C}_{\text{C},\ell}, \mathcal{C}_{\text{F},\ell}, \ell = 1, \ldots, L$ can also be viewed as an ensemble of nested linear codes.

Define the mapping $\phi : \mathbb{Z}_p \to \mathbb{R}$ as

$$\phi(w) \triangleq \gamma p^{-1} w$$

along with the inverse map

$$\bar{\phi}(\kappa) \triangleq [\gamma^{-1} p \kappa] \bmod p \ ,$$

which is only defined over the domain $\gamma p^{-1}\mathbb{Z}$. When applied to vectors, these mappings operate elementwise.

Following (a scaled version[17] of) Construction A, we create the lattices

$$\Lambda_{\text{C},\ell} = \left\{\boldsymbol{\lambda} \in \gamma p^{-1}\mathbb{Z}^n : \bar{\phi}(\boldsymbol{\lambda}) \in \mathcal{C}_{\text{C},\ell}\right\}$$
$$\Lambda_{\text{F},\ell} = \left\{\boldsymbol{\lambda} \in \gamma p^{-1}\mathbb{Z}^n : \bar{\phi}(\boldsymbol{\lambda}) \in \mathcal{C}_{\text{F},\ell}\right\}$$

Note that, by construction, $\boldsymbol{\lambda} \in \Lambda_{\text{C},\ell}$ (or $\Lambda_{\text{F},\ell}$) if and only if $\bar{\phi}(\boldsymbol{\lambda}) \in \mathcal{C}_{\text{C},\ell}$ (or $\mathcal{C}_{\text{F},\ell}$). We will refer to $\bar{\phi}(\boldsymbol{\lambda})$ as the *corresponding linear codeword* of $\boldsymbol{\lambda}$.

We denote the Voronoi regions of $\Lambda_{\text{C},\ell}$ and $\Lambda_{\text{F},\ell}$ by $\mathcal{V}_{\text{C},\ell}$ and $\mathcal{V}_{\text{F},\ell}$, respectively. All $2L$ lattices in this ensemble are nested with respect to the permutation that places the parameters $k_{\text{C},\ell}$ and $k_{\text{F},\ell}$ in increasing order (i.e., they form a nested lattice chain). In particular, since $k_{\text{C},\ell} < k_{\text{F},\ell}$, the nested lattice codebook $\mathcal{L}_\ell$ can be constructed using $\Lambda_{\text{C},\ell}$ as the coarse lattice and $\Lambda_{\text{F},\ell}$ as the fine lattice,

$$\mathcal{L}_\ell \triangleq \Lambda_{\text{F},\ell} \cap \mathcal{V}_{\text{C},\ell} \ .$$

The theorem below restates key existence results from [59] in a form that is convenient for our achievability proofs. At a high level, the theorem guarantees that there exists a generator matrix $\mathbf{G}$, such that, for $\ell = 1, \ldots, L$, the submatrices $\mathbf{G}_{\text{C},\ell}, \mathbf{G}_{\text{F},\ell}$ are full rank and that each resulting nested lattice codebook $\mathcal{L}_\ell$ satisfies its power constraint P_ℓ, tolerates effective noise with variance up to $\sigma^2_{\text{eff},\ell} < P_\ell$, and has rate close to $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.

Theorem 8 ([59, Theorem 2]): Consider any choices of powers $P_\ell > 0$ and effective noise tolerances $0 < \sigma^2_{\text{eff},\ell} < P_\ell$

[14] The field $\mathbb{Z}_p$ is considered here rather than a generic prime-sized finite field since we will use the natural mapping from $\mathbb{Z}_p$ to $\mathbb{Z}$ to lift our codes from the finite field to reals.

[15] These parameter choices are made to simplify the existence proofs. For instance, the prime p is chosen to grow with n so that the channel input distributions will look nearly Gaussian. In practice, one could take p to be relatively small and accept the rate loss associated with p-ary inputs to a Gaussian channel.

[16] For any $m > 3$, Bertrand's Postulate, as proven by Chebyshev [86], states that there exists a prime between m and $2m$.

[17] Construction A was originally proposed in [87] as the vectors of the integer lattice $\mathbb{Z}^n$ whose modulo-p reduction are elements of the linear codebook $\mathbb{C} \subset \mathbb{Z}_p^n$, i.e., $\{\boldsymbol{\lambda} \in \mathbb{Z}^n : [\boldsymbol{\lambda}] \bmod p \in \mathcal{C}\}$.

for $\ell = 1, \ldots, L$. For any $\epsilon > 0$ and n large enough, there is a constant $\alpha > 0$ such that for the choices of $k_{\text{C},\ell}, k_{\text{F},\ell}$ in (20)-(21), there exists a matrix $\mathbf{G} \in \mathbb{Z}_p^{k_{\text{F}} \times n}$, such that, for $\ell = 1, \ldots, L$,

(a) the submatrices $\mathbf{G}_{\text{C},\ell}, \mathbf{G}_{\text{F},\ell}$ are full rank.

(b) the coarse lattices $\Lambda_{\text{C},\ell}$ have second moments close to the power constraint
$$P_\ell - \epsilon < \sigma^2(\Lambda_{\text{C},\ell}) \leq P_\ell \ .$$

(c) the lattices tolerate the prescribed level of effective noise. Specifically, consider any linear mixture of Gaussian and Voronoi-shaped noise of the form $\mathbf{z}_{\text{eff}} = \beta_0 \mathbf{z}_0 + \sum_{\ell=1}^{L} \beta_\ell \mathbf{z}_\ell$ where $\beta_0, \beta_1, \ldots, \beta_L \in \mathbb{R}$, $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_\ell \sim \text{Unif}(\mathcal{V}_{\text{C},\ell})$, and the noise components $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_L$ are independent of each other and $\boldsymbol{\lambda}$. Then, for any fine lattice point $\boldsymbol{\lambda} \in \Lambda_{\text{F},m}$,
$$\mathbb{P}\Big(Q_{\Lambda_{\text{F},m}}(\boldsymbol{\lambda} + \mathbf{z}_{\text{eff}}) \neq \boldsymbol{\lambda}\Big) < \epsilon$$
if $\beta_0^2 + \sum_{\ell=1}^{L} \beta_\ell^2 P_\ell \leq \sigma^2_{\text{eff},m}$. Similarly, for any coarse lattice point $\boldsymbol{\lambda} \in \Lambda_{\text{C},m}$,
$$\mathbb{P}\Big(Q_{\Lambda_{\text{C},m}}(\boldsymbol{\lambda} + \mathbf{z}_{\text{eff}}) \neq \boldsymbol{\lambda}\Big) < \epsilon$$
if $\beta_0^2 + \sum_{\ell=1}^{L} \beta_\ell^2 P_\ell \leq P_m^2$.

(d) the nested lattice codebooks $\mathcal{L}_\ell = \Lambda_{\text{F},\ell} \cap \mathcal{V}_{\text{C},\ell}$ have appropriate rates
$$\frac{1}{n} \log |\mathcal{L}_\ell| = \frac{k_{\text{F},\ell} - k_{\text{C},\ell}}{n} \log p > \frac{1}{2} \log\left(\frac{P_\ell}{\sigma^2_{\text{eff},\ell}}\right) - \epsilon \ .$$

□

Note that (a) is established in the proof of [59, Theorem 1], which is then used to establish [59, Theorem 2] as a corollary.

Remark 15: In Theorem 8, we have only stated lattice properties that are essential for our achievability proofs. In many cases, it can be shown that nested lattices satisfying stronger versions of these properties exist, which in turn could be used to relax some of the assumptions in our problem statement. For instance, one can select lattices that are tuned for non-Gaussian channel noise. See [58]–[60] for more details. ◊

C. Linear Labeling

We now show how this ensemble of nested lattice codebooks can be connected to computing linear combinations of messages over $\mathbb{Z}_p$. Roughly speaking, we would like to map messages to nested lattice codewords so that our desired linear combinations can be directly recovered from appropriate integer-linear combinations of the lattice codewords. The original compute-and-forward framework [1] directly employed the Construction A mapping from a linear code to a lattice code. Subsequent work by Feng *et al.* [14] developed a richer set of algebraic connections as well as guidelines for selecting codes and constellations that are amenable to compute-and-forward. In the process, [14] proposed the concept of a *linear labeling* as an elegant way to map between an algebraic message space and a nested lattice code. We adopt this approach for our expanded framework.

Let $\ell_{\text{min}} = \arg\min_\ell k_{\text{C},\ell}$ and $\ell_{\text{max}} = \arg\max_\ell k_{\text{F},\ell}$. It will be convenient to define $\Lambda_{\text{C}} \triangleq \Lambda_{\text{C},\ell_{\text{min}}}$ and $\Lambda_{\text{F}} \triangleq \Lambda_{\text{F},\ell_{\text{max}}}$ as the coarsest and finest lattices in the ensemble, respectively. It follows that $\Lambda_{\text{C}} \subseteq \Lambda_{\text{C},\ell} \subset \Lambda_{\text{F},\ell} \subseteq \Lambda_{\text{F}}$ for $\ell = 1, \ldots, L$. Recall that $k_{\text{C}} \triangleq \min_\ell k_{\text{C},\ell}$, $k_{\text{F}} \triangleq \max_\ell k_{\text{F},\ell}$, and $k \triangleq k_{\text{F}} - k_{\text{C}}$.

Definition 11: A mapping $\varphi : \Lambda_{\text{F}} \to \mathbb{Z}_p^k$ is called a *linear labeling* if it satisfies the following two properties:

(a) A lattice point $\boldsymbol{\lambda}$ belongs to $\Lambda_{\text{F},\ell}$ if and only if the last $k_{\text{F}} - k_{\text{F},\ell}$ components of its label $\varphi(\boldsymbol{\lambda})$ are equal to 0. Similarly, a lattice point $\boldsymbol{\lambda}$ belongs to $\Lambda_{\text{C},\ell}$ if and only if the last $k_{\text{F}} - k_{\text{C},\ell}$ components of its label $\varphi(\boldsymbol{\lambda})$ are equal to 0.

(b) For all $a_\ell \in \mathbb{Z}$ and $\boldsymbol{\lambda}_\ell \in \Lambda_{\text{F}}$, we have that
$$\varphi\left(\sum_{\ell=1}^{L} a_\ell \boldsymbol{\lambda}_\ell\right) = \bigoplus_{\ell=1}^{L} q_\ell \, \varphi(\boldsymbol{\lambda}_\ell)$$
where $q_\ell = [a_\ell] \bmod p$.

◊

Our proposed linear labeling stems directly from the fact that, for any $\boldsymbol{\lambda} \in \Lambda_{\text{F}}$, the corresponding linear codeword can be expressed as $\bar{\phi}(\boldsymbol{\lambda}) = \mathbf{G}^{\mathsf{T}} \mathbf{v}$ for some vector $\mathbf{v} \in \mathbb{Z}_p^{k_{\text{F}}}$. Assuming that $\mathbf{G}$ is full rank, then this vector is unique.

Theorem 9: Assume that $\mathbf{G}$ is full rank. Let $\varphi : \Lambda_{\text{F}} \to \mathbb{Z}_p^k$ be the function that maps each $\boldsymbol{\lambda} \in \Lambda_{\text{F}}$ to the vector $\varphi(\boldsymbol{\lambda})$ that consists of the last k components of the unique vector $\mathbf{v}$ satisfying $\bar{\phi}(\boldsymbol{\lambda}) = \mathbf{G}^{\mathsf{T}} \mathbf{v}$. Then, φ is a linear labeling. □

See Appendix G for a proof. We have depicted Theorem 9 through signal levels in Figure 7.

It will also be useful to have an explicit inverse $\bar{\varphi} : \mathbb{Z}_p^k \to \Lambda_{\text{F}}$ for the linear labeling φ from Theorem 9. Specifically, define
$$\bar{\varphi}(\mathbf{w}) \triangleq \phi\left(\mathbf{G}^{\mathsf{T}} \begin{bmatrix} \mathbf{0}_{k_{\text{C}}} \\ \mathbf{w} \end{bmatrix}\right) .$$
It follows that $\varphi\big(\bar{\varphi}(\mathbf{w})\big) = \mathbf{w}$.

We now have all the ingredients we need to construct coding schemes for compute-and-forward. The next two sections develop coding schemes for parallel and successive computation, respectively.

V. Parallel Computation Achievability: Proof of Theorem 1

We now provide a detailed description of the encoding and decoding strategies used to achieve the parallel computation rate region from Theorem 1. Notice that, for a given integer coefficient matrix $\mathbf{A}$, we determine the computation rate region $\mathcal{R}_{\text{comp}}^{(\text{para})}(\mathbf{H}, \mathbf{A})$ by taking the union over all integer matrices $\tilde{\mathbf{A}}$ whose rowspan contains that of $\mathbf{A}$. This has a clear operational meaning within our scheme: the receiver recovers linear combinations with integer coefficient matrix $\tilde{\mathbf{A}}$ and then solves these for the desired linear combinations with integer coefficient matrix $\mathbf{A}$. As a first step, we will show how to directly recover linear combinations with integer coefficient matrix $\mathbf{A}$ (for notational simplicity,) and derive conditions under which the probability of error can be driven to zero in Lemma 10. This will serve as a building block in the proof of Theorem 1.

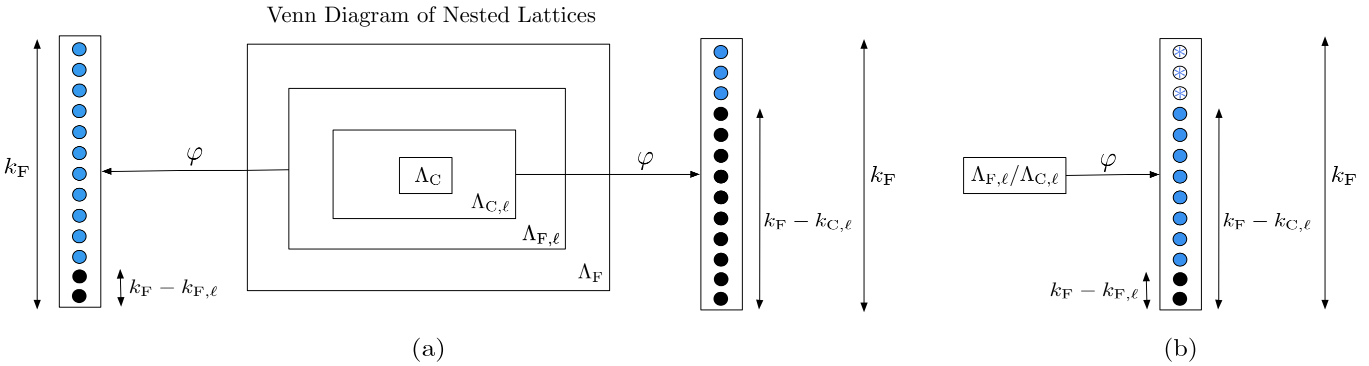
Venn Diagram of Nested Lattices
k_F
φ
$k_\mathrm{F}-k_{\mathrm{F},\ell}$
Λ_C
$\Lambda_{\mathrm{C},\ell}$
$\Lambda_{\mathrm{F},\ell}$
Λ_F
φ
$k_\mathrm{F}-k_{\mathrm{C},\ell}$
k_F
(a)
$\Lambda_{\mathrm{F},\ell}/\Lambda_{\mathrm{C},\ell}$
φ
$k_\mathrm{F}-k_{\mathrm{F},\ell}$
$k_\mathrm{F}-k_{\mathrm{C},\ell}$
k_F
(b)

Fig. 7. (a) Depiction of the linear labeling φ of Theorem 9 via signal levels for a user ℓ with $k_\mathrm{F}=13$, $k_{\mathrm{F},\ell}=11$, and $k_{\mathrm{C},\ell}=3$. We use black circles to represent zeros and lightly shaded (blue) circles to represent occupied symbols. On the left-hand side, we see that the linear labeling of a lattice point from $\Lambda_{\mathrm{F},\ell}$ must have zeros in its last $k_\mathrm{F}-k_{\mathrm{F},\ell}$ entries. Similarly, on the right-hand side, we see that the linear labeling of a lattice point from $\Lambda_{\mathrm{C},\ell}$ must have zeros in its last $k_\mathrm{F}-k_{\mathrm{C},\ell}$ entries. (b) Depiction of the linear labeling φ of Theorem 9 of an element of $\Lambda_{\mathrm{F},\ell}/\Lambda_{\mathrm{C},\ell}$. The $*$ elements represent "don't care" entries; for a given choice of signal levels represented by the blue circles, different realizations of the "don't care" levels correspond to different representatives of the same coset.

We begin with a high-level description of the encoding steps used to map the finite field messages onto dithered lattice codewords as well as the decoding steps used to estimate integer-linear combinations of lattice codewords, which are then mapped back to linear combinations of the messages. Take any choice of rates $R_1,\dots,R_L$ and parameter $\epsilon>0$, and apply Theorem 8 to select good nested lattices. Afterwards, apply Theorem 9 to obtain a linear labeling φ and its inverse $\bar{\varphi}$. The encoding and decoding process will make use of random[18] dither vectors that are generated independently and uniformly over the Voronoi regions of the coarse lattices, $\mathbf{d}_\ell\sim\mathrm{Unif}(\mathcal{V}_{\mathrm{C},\ell})$.

The ℓth encoder begins by adding $k_{\mathrm{C},\ell}-k_\mathrm{C}$ leading zeros and $k_\mathrm{F}-k_{\mathrm{F},\ell}$ trailing zeros to its message $\mathbf{w}_\ell\in\mathbb{Z}_p^{k_{\mathrm{F},\ell}-k_{\mathrm{C},\ell}}$. The resulting length-$k$ vector is mapped onto a lattice point $\boldsymbol{\lambda}_\ell\in\mathcal{L}_\ell$ using the inverse $\bar{\varphi}$ of the linear labeling followed by taking modulo $\Lambda_{c,\ell}$. Afterwards, the encoder dithers $\boldsymbol{\lambda}_\ell$ using $\mathbf{d}_\ell$ to obtain its channel input $\mathbf{x}_\ell$. These encoding operations are illustrated in Figure 8 and formally written out in (22). Note that encoding does not depend on either the channel matrix $\mathbf{H}$ or the choice of integer coefficient vector $\mathbf{a}_m$.

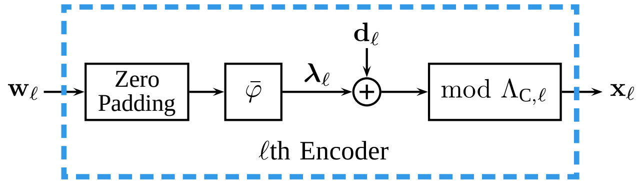

Fig. 8. Block diagram for the ℓth encoder for both parallel computation and successive computation.

Encoding:

$$\boldsymbol{\lambda}_\ell=\left[\bar{\varphi}\left(\begin{bmatrix}\mathbf{0}_{k_{\mathrm{C},\ell}-k_\mathrm{C}}\\ \mathbf{w}_\ell\\ \mathbf{0}_{k_\mathrm{F}-k_{\mathrm{F},\ell}}\end{bmatrix}\right)\right]\bmod\Lambda_{\mathrm{C},\ell}\tag{22a}$$

$$\mathbf{x}_\ell=[\boldsymbol{\lambda}_\ell+\mathbf{d}_\ell]\bmod\Lambda_{\mathrm{C},\ell}\tag{22b}$$

To recover the linear combination $\mathbf{u}_m$ with integer coefficient vector $\mathbf{a}_m$, the receiver first applies the equalization vector $\mathbf{b}_m\in\mathbb{R}^{N_\mathrm{r}}$ to its channel observation $\mathbf{Y}$, to obtain the effective channel output $\tilde{\mathbf{y}}_m$. The receiver then attempts to recover the integer-linear combination

$$\boldsymbol{\mu}_m=\left[\sum_{\ell=1}^{L}a_{m,\ell}\tilde{\boldsymbol{\lambda}}_\ell\right]\bmod\Lambda_\mathrm{C}\tag{23}$$

where

$$\tilde{\boldsymbol{\lambda}}_\ell\triangleq\boldsymbol{\lambda}_\ell-Q_{\Lambda_{\mathrm{C},\ell}}(\boldsymbol{\lambda}_\ell+\mathbf{d}_\ell)\tag{24}$$

is a lattice point in the same coset $\Lambda_{\mathrm{F},\ell}/\Lambda_{\mathrm{C},\ell}$ as the ℓth user's lattice codeword $\boldsymbol{\lambda}_\ell$. An estimate $\hat{\boldsymbol{\mu}}_m$ of $\boldsymbol{\mu}_m$ is obtained by first subtracting an integer-linear combination of the dithers from $\tilde{\mathbf{y}}_m$, then quantizing onto the fine lattice $\Lambda_{\mathrm{F},\theta(m)}$ where

$$\theta(m)\triangleq\arg\max\left\{k_{\mathrm{F},\ell}:\ell\in\{1,\dots,L\}\text{ s.t. }[a_{m,\ell}]\bmod p\neq 0\right\}$$

denotes the index of the finest lattice amongst those that participate in the integer-linear combination $\boldsymbol{\mu}_m$, and finally taking mod Λ_C. To make an estimate $\hat{\mathbf{u}}_m$ of the desired linear combination $\mathbf{u}_m$, the receiver simply applies the linear labeling φ to $\hat{\boldsymbol{\mu}}_m$. These decoding operations are illustrated in Figure 9 and formally written out in (25).

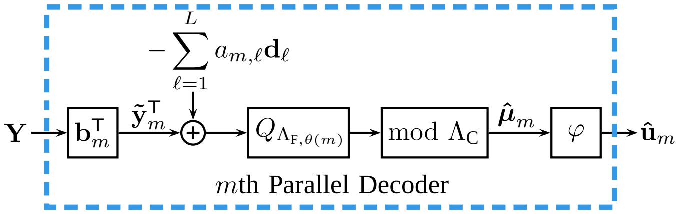
$-\sum_{\ell=1}^{L}a_{m,\ell}\mathbf{d}_\ell$
$\mathbf{Y}$
$\mathbf{b}_m^\mathsf{T}$
$\tilde{\mathbf{y}}_m^\mathsf{T}$
$Q_{\Lambda_{\mathrm{F},\theta(m)}}$
mod Λ_C
$\hat{\boldsymbol{\mu}}_m$
φ
$\hat{\mathbf{u}}_m$
mth Parallel Decoder

Fig. 9. Block diagram for the mth decoder for parallel computation.

Decoding:

$$\tilde{\mathbf{y}}_m^\mathsf{T}=\mathbf{b}_m^\mathsf{T}\mathbf{Y}\tag{25a}$$

$$\hat{\boldsymbol{\mu}}_m=\left[Q_{\Lambda_{\mathrm{F},\theta(m)}}\left(\tilde{\mathbf{y}}_m-\sum_{\ell=1}^{L}a_{m,\ell}\,\mathbf{d}_\ell\right)\right]\bmod\Lambda_\mathrm{C}\tag{25b}$$

$$\hat{\mathbf{u}}_m=\varphi\big(\hat{\boldsymbol{\mu}}_m\big)\tag{25c}$$

[18] The use of random dither vectors should not be viewed as common randomness, but rather as part of the random coding proof. In Appendix H, we will show that it suffices to use fixed dither vectors.

The following lemma describes rates that are attainable for the coding strategy above.

Lemma 10: Consider any choice of rates $R_1, \dots, R_L$ and parameter $\zeta > 0$ and assume that the transmitters and receiver employ the encoding and decoding operations from (22)-(25). For n and prime p large enough, there exists a generator matrix $\mathbf{G} \in \mathbb{Z}_p^{k_{\text{F}} \times n}$ and corresponding nested lattice codes $\mathcal{L}_1, \dots, \mathcal{L}_L$ with rates at least $R_1 - \zeta, \dots, R_L - \zeta$, respectively, such that, for any choice of channel matrix $\mathbf{H} \in \mathbb{R}^{N_{\text{r}} \times L}$ and integer matrix $\mathbf{A} \in \mathbb{Z}^{L \times L}$, the receiver can recover the linear combinations $\mathbf{u}_1, \dots, \mathbf{u}_L$ with probability of error at most ζ so long as

$$R_\ell < \frac{1}{2} \log^+ \left(\frac{P_\ell}{\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m)} \right) \tag{26}$$

for all (m, ℓ) such that $[a_{m,\ell}] \bmod p \neq 0$ for some choice of equalization vectors $\mathbf{b}_1, \dots, \mathbf{b}_L \in \mathbb{R}^{N_{\text{r}}}$. □

Proof: First, we will select an ensemble of good nested lattice codebooks. Set $\sigma^2_{\text{eff},\ell} = P_\ell 2^{-2R_\ell}$ for $\ell = 1, \dots, L$. Invoke Theorem 8 with these $\sigma^2_{\text{eff},1}, \dots, \sigma^2_{\text{eff},L}$ and parameter $\epsilon = \zeta/L$ to obtain a generator matrix $\mathbf{G} \in \mathbb{Z}_p^{k_{\text{F}} \times n}$ and associated nested lattice codebooks $\mathcal{L}_1, \dots, \mathcal{L}_L$ with rates at least $R_1 - \epsilon, \dots, R_L - \epsilon$, respectively. Use Theorem 9 to obtain a linear labeling $\varphi : \Lambda_{\text{F}} \to \mathbb{Z}_p^k$ and its inverse $\bar{\varphi} : \mathbb{Z}_p^k \to \Lambda_{\text{F}}$.

For $\ell = 1, \dots, L$, generate an independent random dither vector according to $\mathbf{d}_\ell \sim \text{Unif}(\mathcal{V}_{\text{C},\ell})$ and make it available to the ℓth transmitter and the receiver. Each transmitter employs (22) to generate its lattice codeword $\boldsymbol{\lambda}_\ell$ and channel input $\mathbf{x}_\ell$. From the Crypto Lemma, we know that $\mathbf{x}_\ell$ is independent of $\boldsymbol{\lambda}_\ell$ and uniform over $\mathcal{V}_{\text{C},\ell}$. Therefore, it follows from Theorem 8(b) that each transmitter satisfies its power constraint (1).

For $m = 1, \dots, L$, the receiver selects an equalization vector $\mathbf{b}_m \in \mathbb{R}^{N_{\text{r}}}$ and generates an effective channel output $\tilde{\mathbf{y}}_m$ via (25a). After subtracting an integer combination of the dither vectors, we obtain an integer combination of lattice codewords plus effective noise:

$$\begin{aligned}
&\tilde{\mathbf{y}}_m - \sum_{\ell=1}^{L} a_{m,\ell} \, \mathbf{d}_\ell \\
&= \sum_{\ell=1}^{L} a_{m,\ell} (\mathbf{x}_\ell - \mathbf{d}_\ell) + \mathbf{z}_{\text{para},m} \\
&= \sum_{\ell=1}^{L} a_{m,\ell} \big(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell - Q_{\Lambda_{\text{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell) - \mathbf{d}_\ell \big) + \mathbf{z}_{\text{para},m} \\
&= \sum_{\ell=1}^{L} a_{m,\ell} \tilde{\boldsymbol{\lambda}}_\ell + \mathbf{z}_{\text{para},m}
\end{aligned}$$

where

$$\mathbf{z}^{\mathsf{T}}_{\text{para},m} \triangleq (\mathbf{b}_m^{\mathsf{T}} \mathbf{H} - \mathbf{a}_m^{\mathsf{T}}) \mathbf{X} + \mathbf{b}_m^{\mathsf{T}} \mathbf{Z} \ .$$

The linear label of $\tilde{\boldsymbol{\lambda}}_\ell$ satisfies

$$\varphi(\tilde{\boldsymbol{\lambda}}_\ell) = \varphi(\boldsymbol{\lambda}_\ell) \ominus \varphi\big(Q_{\Lambda_{\text{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell)\big)$$

where $\ominus$ denotes subtraction over $\mathbb{Z}_p$. By Definition 11(a), since $Q_{\Lambda_{\text{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell) \in \Lambda_{\text{C},\ell}$, the last $k_{\text{F}} - k_{\text{C},\ell}$ components of the label $\varphi\big(Q_{\Lambda_{\text{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell)\big)$ are zero. Therefore, $\varphi(\tilde{\boldsymbol{\lambda}}_\ell)$ agrees with $\varphi(\boldsymbol{\lambda}_\ell)$ on its last $k_{\text{F}} - k_{\text{C},\ell}$ components, i.e., it may not agree on the first $k - (k_{\text{F}} - k_{\text{C},\ell}) = k_{\text{F}} - k_{\text{C}} - k_{\text{F}} + k_{\text{C},\ell} = k_{\text{C},\ell} - k_{\text{C}}$ components. This implies that $\varphi(\tilde{\boldsymbol{\lambda}}_\ell) \in [\![\mathbf{w}_\ell]\!]$ as defined in (6). By Definition 11(b), $\varphi\big(\sum_\ell a_{m,\ell} \, \tilde{\boldsymbol{\lambda}}_\ell\big) = \bigoplus_\ell q_{m,\ell} \, \varphi(\tilde{\boldsymbol{\lambda}}_\ell)$ where $q_{m,\ell} = [a_{m,\ell}] \bmod p$ so the linear label can be viewed as a linear combination $\mathbf{u}_m = \varphi\big(\sum_\ell a_{m,\ell} \, \tilde{\boldsymbol{\lambda}}_\ell\big)$ with integer coefficient vector $\mathbf{a}_m$. This also implies, via Definition 11(a), that $\sum_\ell a_{m,\ell} \tilde{\boldsymbol{\lambda}}_\ell \in \Lambda_{\text{F},\theta(m)}$, which will be useful in the next step.

Applying (25b), the receiver makes an estimate $\hat{\boldsymbol{\mu}}_m$ of the integer-linear combination $\boldsymbol{\mu}_m$. By Theorem 8(c), we have that $\mathbb{P}(\hat{\boldsymbol{\mu}}_m \neq \boldsymbol{\mu}_m) < \epsilon$ if

$$\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m) < \sigma^2_{\text{eff},\theta(m)} \tag{27}$$

where $\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m)$ is defined in (9). Afterwards, the receiver applies the linear labeling as in (25c) to obtain its estimate $\hat{\mathbf{u}}_m$. If $\hat{\boldsymbol{\mu}}_m = \boldsymbol{\mu}_m$, then $\hat{\mathbf{u}}_m = \mathbf{u}_m$ since $\varphi(\boldsymbol{\mu}_m) = \varphi\big(\sum_\ell a_{m,\ell} \tilde{\boldsymbol{\lambda}}_\ell\big) \ominus \varphi\big(Q_{\Lambda_{\text{C}}}\big(\sum_\ell a_{m,\ell} \tilde{\boldsymbol{\lambda}}_\ell\big)\big)$ and the second term is equal to $\mathbf{0}_k$ since it is a linear labeling of an element of Λ_{C} (i.e., the last $k_{\text{F}} - k_{\text{C}} = k$ components of its label are zero).

Note that the condition (27) is equivalent to insisting that $R_\ell < \frac{1}{2} \log^+ \big(P_\ell / \sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m)\big)$ for all m and ℓ such that $[a_{m,\ell}] \bmod p \neq 0$. Applying the union bound, we get that $\mathbb{P}\big(\cup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\big) < L\epsilon = \zeta$. Finally, it is argued in Appendix H that it suffices to use fixed dither vectors. ■

We are now ready to prove Theorem 1.

Proof of Theorem 1: Choose rates $R_1, \dots, R_L$ and select nested lattice codebooks via Lemma 10. Each transmitter employs the encoding strategy from Lemma 10 (which does not depend on the channel matrix $\mathbf{H}$ nor the integer matrix $\mathbf{A}$). For a given channel matrix $\mathbf{H}$, say that the receiver wishes to decode linear combinations with integer coefficient matrix $\mathbf{A}$ where $(R_1, \dots, R_L) \in \mathcal{R}^{(\text{para})}_{\text{comp}}(\mathbf{H}, \mathbf{A})$. This implies that there exists some integer matrix $\tilde{\mathbf{A}}$ satisfying $\text{rowspan}(\tilde{\mathbf{A}}) \subseteq \text{rowspan}(\mathbf{A})$ and $(R_1, \dots, R_L) \in \mathcal{R}_{\text{para}}(\mathbf{H}, \tilde{\mathbf{A}})$. The receiver applies the decoding strategy from Lemma 10 with integer matrix $\tilde{\mathbf{A}}$ and optimal equalization vectors $\mathbf{b}_{\text{opt},m}$ chosen via Lemma 2. Recall from (10) that $\sigma^2_{\text{para}}(\mathbf{H}, \tilde{\mathbf{a}}_m) \triangleq \sigma^2_{\text{para}}(\mathbf{H}, \tilde{\mathbf{a}}_m, \mathbf{b}_{\text{opt},m})$ so (26) matches the rate constraints from $\mathcal{R}_{\text{para}}(\mathbf{H}, \tilde{\mathbf{A}})$. ■

VI. Successive Computation Achievability: Proof of Theorem 2

In this section, we show how to improve the decoding process using successive cancellation, culminating in a proof of Theorem 2. The successive computation rate region $\mathcal{R}^{(\text{para})}_{\text{comp}}(\mathbf{H}, \mathbf{A})$ involves a union over all integer matrices $\tilde{\mathbf{A}}$ whose rowspan contains that of $\mathbf{A}$ as well as a union over all admissible mappings $\mathcal{M}(\tilde{\mathbf{A}})$ for each $\tilde{\mathbf{A}}$. As before, the union over integer matrices means that the receiver can first recover linear combinations with integer coefficient matrix $\tilde{\mathbf{A}}$, and then solve these for the desired linear combinations with integer coefficient matrix $\mathbf{A}$. Each admissible mapping $\mathcal{I} \in \mathcal{M}(\tilde{\mathbf{A}})$ corresponds to a specific successive cancellation order for the codewords, as we will explain in detail below. We will begin by showing how to directly recover linear combinations with coefficient matrix $\mathbf{A}$ and admissible mapping $\mathcal{I}$ (for notational simplicity), and, in Lemma 13, state conditions under which the probability of error can be made to vanish in the blocklength. Afterwards, we will use this lemma to prove Theorem 2. Note that the decoding order for linear combinations is fixed to be lexicographic for notational convenience; other orders can be reached by exchanging rows of $\mathbf{A}$ (which is taken care of by the union over integer matrices that include the rowspan of $\mathbf{A}$).

We start with a high-level overview of our encoding and decoding process. As in Section V, we fix rates $R_1, \ldots, R_L$ and parameter $\epsilon > 0$ and then invoke Theorems 8 and 9 to select good nested lattices as well as a linear labeling φ and its inverse $\bar{\varphi}$. Next, generate random dither vectors independently and uniformly over the Voronoi regions of the coarse lattices, $\mathbf{d}_\ell \sim \mathrm{Unif}(\mathcal{V}_{\mathrm{C},\ell})$.

The encoding process is identical to that in the parallel computation case. We summarize the operations in (28) and refer the reader to Section V for a detailed discussion and Figure 8 for a block diagram.

Encoding:

$$\boldsymbol{\lambda}_\ell = \left[\bar{\varphi}\left(\begin{bmatrix} \mathbf{0}_{k_{\mathrm{C},\ell}-k_{\mathrm{C}}} \\ \mathbf{w}_\ell \\ \mathbf{0}_{k_{\mathrm{F}}-k_{\mathrm{F},\ell}} \end{bmatrix} \right) \right] \bmod \Lambda_{\mathrm{C},\ell} \tag{28a}$$

$$\mathbf{x}_\ell = [\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathrm{C},\ell} \tag{28b}$$

The first decoding step (29a)-(29c) is quite similar to that of parallel computation (since there is not yet any side information to exploit). The decoder first recovers an integer-linear combination of lattice codewords $\boldsymbol{\mu}_1$ as defined in (23) and then applies the linear labeling. Let

$$\vartheta(m) \triangleq \arg\max \left\{ k_{\mathrm{F},\ell} : \ell \in \{1, \ldots, L\} \text{ s.t. } (m, \ell) \in \mathcal{I} \right\}$$

denote the index of finest lattice that will participate in the mth integer-linear combination under the admissible mapping $\mathcal{I}$. Note that $\vartheta(1) = \theta(1)$ since no successive cancellation takes place in the first round.

Decoding, $\mathbf{m} = \mathbf{1}$:

$$\tilde{\mathbf{y}}_1^{\mathsf{T}} = \mathbf{b}_1^{\mathsf{T}} \mathbf{Y} \tag{29a}$$

$$\hat{\boldsymbol{\mu}}_1 = \left[Q_{\Lambda_{\mathrm{F},\vartheta(1)}} \left(\tilde{\mathbf{y}}_1 - \sum_{\ell=1}^{L} a_{1,\ell}\, \mathbf{d}_\ell \right) \right] \bmod \Lambda_{\mathrm{C}} \tag{29b}$$

$$\hat{\mathbf{u}}_1 = \varphi(\hat{\boldsymbol{\mu}}_1) \tag{29c}$$

$$\hat{\boldsymbol{\chi}}_1 = \left[\hat{\boldsymbol{\mu}}_1 + \sum_{\ell=1}^{L} a_{1,\ell}\, \mathbf{d}_\ell \right] \bmod \Lambda_{\mathrm{C}} \tag{29d}$$

$$\hat{\mathbf{s}}_1 = Q_{\Lambda_{\mathrm{C}}}(\tilde{\mathbf{y}}_1 - \hat{\boldsymbol{\chi}}_1) + \hat{\boldsymbol{\chi}}_1 \tag{29e}$$

After making its estimate $\hat{\mathbf{u}}_1$, the decoder attempts to reconstruct the integer-linear combination of channel inputs $\mathbf{a}_1^{\mathsf{T}}\mathbf{X}$ from $\hat{\boldsymbol{\mu}}_1$ and $\tilde{\mathbf{y}}_1$ in (29d)-(29e). The following lemma characterizes when this process succeeds.

Lemma 11: Assume a receiver has access to an observation of the form $\tilde{\mathbf{y}}^{\mathsf{T}} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + \mathbf{z}_{\text{eff}}^{\mathsf{T}}$ where $\mathbf{a} \in \mathbb{Z}^L$ and $\mathbf{z}_{\text{eff}} \in \mathcal{V}_{\mathrm{C}}$, dithers $\mathbf{d}_1, \ldots, \mathbf{d}_L$, and the integer-linear combination $\boldsymbol{\mu} = \left[\sum_\ell a_\ell \tilde{\boldsymbol{\lambda}}_\ell\right] \bmod \Lambda_{\mathrm{C}}$ where a_ℓ is the ℓth entry of $\mathbf{a}$ and $\tilde{\boldsymbol{\lambda}}_\ell = \boldsymbol{\lambda}_\ell - Q_{\Lambda_{\mathrm{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell)$. Then, by calculating

$$\boldsymbol{\chi} = \left[\boldsymbol{\mu} + \sum_{\ell=1}^{L} a_\ell\, \mathbf{d}_\ell \right] \bmod \Lambda_{\mathrm{C}}$$

$$\mathbf{s} = Q_{\Lambda_{\mathrm{C}}}(\tilde{\mathbf{y}} - \boldsymbol{\chi}) + \boldsymbol{\chi} \ ,$$

the receiver can obtain the integer-linear combination of the channel inputs, $\mathbf{s}^{\mathsf{T}} = \mathbf{a}^{\mathsf{T}}\mathbf{X}$.

Proof: By the distributive law,

$$\begin{aligned} \boldsymbol{\chi} &= \left[\sum_{\ell=1}^{L} a_\ell (\tilde{\boldsymbol{\lambda}}_\ell + \mathbf{d}_\ell) \right] \bmod \Lambda_{\mathrm{C}} \\ &= \left[\sum_{\ell=1}^{L} a_\ell \Big(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell - Q_{\Lambda_{\mathrm{C},\ell}}(\boldsymbol{\lambda}_\ell + \mathbf{d}_\ell) \Big) \right] \bmod \Lambda_{\mathrm{C}} \\ &= \left[\sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell \right] \bmod \Lambda_{\mathrm{C}} \ . \end{aligned} \tag{30}$$

Therefore,

$$\begin{aligned} \mathbf{s}^{\mathsf{T}} &= Q_{\Lambda_{\mathrm{C}}}(\tilde{\mathbf{y}}^{\mathsf{T}} - \boldsymbol{\chi}^{\mathsf{T}}) + \boldsymbol{\chi}^{\mathsf{T}} \\ &= Q_{\Lambda_{\mathrm{C}}}\Big(\mathbf{a}^{\mathsf{T}}\mathbf{X} + \mathbf{z}_{\text{eff}}^{\mathsf{T}} - \big[\mathbf{a}^{\mathsf{T}}\mathbf{X}\big] \bmod \Lambda_{\mathrm{C}} \Big) + \boldsymbol{\chi}^{\mathsf{T}} \\ &= Q_{\Lambda_{\mathrm{C}}}\Big(Q_{\Lambda_{\mathrm{C}}}(\mathbf{a}^{\mathsf{T}}\mathbf{X}) + \mathbf{z}_{\text{eff}}^{\mathsf{T}} \Big) + \boldsymbol{\chi}^{\mathsf{T}} \\ &\overset{(i)}{=} Q_{\Lambda_{\mathrm{C}}}(\mathbf{a}^{\mathsf{T}}\mathbf{X}) + \big[\mathbf{a}^{\mathsf{T}}\mathbf{X}\big] \bmod \Lambda_{\mathrm{C}} \\ &= \mathbf{a}^{\mathsf{T}}\mathbf{X} \end{aligned}$$

where (i) uses the fact that $\mathbf{z}_{\text{eff}} \in \mathcal{V}_{\mathrm{C}}$ as well as (30). ■

Thus, if $\hat{\boldsymbol{\mu}}_1 = \boldsymbol{\mu}_1$, Lemma 11 will allow us to argue that $\hat{\mathbf{s}}_1^{\mathsf{T}} = \mathbf{a}_1^{\mathsf{T}}\mathbf{X}$, which can be used for *successive computation* as proposed by [16], i.e., creating better effective channels for subsequent linear combinations. In general, at the mth decoding step, we will have access to $\mathbf{A}_{m-1}\mathbf{X}$ where $\mathbf{A}_{m-1}$ is the submatrix consisting of the first $m-1$ rows of $\mathbf{A}$, assuming all previous decoding steps are correct.

The second ingredient in our decoding process is *algebraic successive cancellation* as proposed by [9]. The main idea is that, at decoding step m, it is possible to use linear combinations from steps 1 through $m-1$, to cancel out some of the codewords participating in the integer-linear combination $\boldsymbol{\mu}_m$ without changing the effective noise variance. This in turn reduces the noise tolerance constraints placed on the fine lattices associated with the codewords and increases the overall rate region. Before we proceed, we need the following lemma that connects the definition of an admissible mapping to the existence of a matrix over $\mathbb{Z}_p$ that can be used for algebraic successive cancellation.

Lemma 12: Let $\mathcal{I}$ be an admissible mapping for $\mathbf{A} \in \mathbb{Z}^{L\times L}$. For prime p large enough, there exists a lower unitriangular matrix $\bar{\mathbf{L}} \in \mathbb{Z}_p^{L\times L}$ such that, the (m,ℓ)th entry of $\bar{\mathbf{A}} = [\bar{\mathbf{L}}\mathbf{A}] \bmod p$ is equal to zero (i.e., $\bar{a}_{m,\ell} = 0$) for all $(m,\ell) \neq \mathcal{I}$. Furthermore, $\bar{\mathbf{L}}$ has a lower triangular inverse $\bar{\mathbf{L}}^{(\text{inv})} \in \mathbb{Z}_p^{L\times L}$ satisfying $[\mathbf{A}] \bmod p = [\bar{\mathbf{L}}^{(\text{inv})}\bar{\mathbf{A}}] \bmod p$.

Proof: By Definition 5, since $\mathcal{I}$ is an admissible mapping, there exists a real-valued, lower unitriangular matrix

$\mathbf{L} \in \mathbb{R}^{L \times L}$ such that the (m, ℓ)th entry of $\mathbf{LA}$ is equal to zero for all $(m, \ell) \neq \mathcal{I}$. It follows from [9, Appendix A] that, for p large enough, there exists a lower unitriangular matrix $\bar{\mathbf{L}} \in \mathbb{Z}_p^{L \times L}$ satisfying the same criterion for $\bar{\mathbf{A}} = [\bar{\mathbf{L}}\mathbf{A}] \bmod p$. Finally, since $\bar{\mathbf{L}}$ is lower unitriangular, it has a lower unitriangular inverse over $\mathbb{Z}_p$. ■

An immediate consequence of Lemma 12 is that we can use preceding linear combinations to eliminate lattice codewords according to the admissible mapping,

$$\begin{aligned}\boldsymbol{\nu}_m &= \left[\boldsymbol{\mu}_m + \sum_{i=1}^{m-1} \bar{l}_{m,i}\boldsymbol{\mu}_i\right] \bmod \Lambda_{\mathrm{C}} \\ &= \left[\sum_{\ell=1}^{L} \bar{a}_{m,\ell}\, \tilde{\boldsymbol{\lambda}}_\ell\right] \bmod \Lambda_{\mathrm{C}} \qquad (31)\end{aligned}$$

where $\bar{l}_{m,i}$ is the (m, i)th entry of $\bar{\mathbf{L}}$ chosen via Lemma 12 and $\bar{a}_{m,\ell}$ is the (m, ℓ) entry of $\bar{\mathbf{A}} = [\bar{\mathbf{L}}\mathbf{A}] \bmod p$. Similarly, using the inverse $\bar{\mathbf{L}}^{(\mathrm{inv})}$ of $\bar{\mathbf{L}}$, we can return to the original integer-linear combinations,

$$\left[\boldsymbol{\nu}_m + \sum_{i=1}^{m-1} \bar{l}^{(\mathrm{inv})}_{m,i} \boldsymbol{\nu}_i\right] \bmod \Lambda_{\mathrm{C}} = \boldsymbol{\mu}_m \qquad (32)$$

where $\bar{l}^{(\mathrm{inv})}_{m,i}$ is the (m, i)th entry of $\bar{\mathbf{L}}^{(\mathrm{inv})}$.

Overall, for $m \geq 2$, the mth successive decoding step begins by assembling the estimates of the integer-linear combinations of channel inputs from the previous $m - 1$ decoding steps into a matrix $\hat{\mathbf{S}}_{m-1}$. Assuming prior steps are correct, we have that $\hat{\mathbf{S}}_{m-1} = \mathbf{A}_{m-1}\mathbf{X}$. It then applies equalization vectors $\mathbf{b}_m \in \mathbb{R}^{N_{\mathrm{r}}}$ and $\mathbf{c}_m \in \mathbb{R}^{m-1}$ to its observation $\mathbf{Y}$ and side information $\hat{\mathbf{S}}_{m-1}$, respectively, and adds the results together to form its effective channel observation $\tilde{\mathbf{y}}_m$. Next, it applies algebraic successive cancellation, removes the dithers, and quantizes onto the appropriate fine lattice. After quantizing, it reverses the algebraic successive cancellation to obtain an estimate $\hat{\boldsymbol{\mu}}_m$ of the integer-linear combination $\boldsymbol{\mu}_m$ from (23). Finally, the decoder uses the linear labeling to make its estimate of the desired linear combination and follows the steps from Lemma 11 to estimate the corresponding integer-linear combination of the channel inputs. These operations are depicted in Figure 10 and formally expressed in (33).

Decoding, m $\geq$ 2:

$$\hat{\mathbf{S}}_{m-1} \triangleq [\hat{\mathbf{s}}_1 \ \cdots \ \hat{\mathbf{s}}_{m-1}]^{\mathsf{T}} \qquad (33a)$$

$$\tilde{\mathbf{y}}_m = \mathbf{b}_m^{\mathsf{T}}\mathbf{Y} + \mathbf{c}_m^{\mathsf{T}}\hat{\mathbf{S}}_{m-1} \qquad (33b)$$

$$\hat{\boldsymbol{\nu}}_m = \left[Q_{\Lambda_{\mathrm{F},\vartheta(m)}}\left(\tilde{\mathbf{y}}_m + \sum_{i=1}^{m-1} \bar{l}_{m,i}\hat{\boldsymbol{\mu}}_i - \sum_{\ell=1}^{L} a_{m,\ell}\,\mathbf{d}_\ell\right)\right] \bmod \Lambda_{\mathrm{C}} \qquad (33c)$$

$$\hat{\boldsymbol{\mu}}_m = \left[\hat{\boldsymbol{\nu}}_m + \sum_{i=1}^{m-1} \bar{l}^{(\mathrm{inv})}_{m,i}\hat{\boldsymbol{\nu}}_i\right] \bmod \Lambda_{\mathrm{C}} \qquad (33d)$$

$$\hat{\mathbf{u}}_m = \varphi(\hat{\boldsymbol{\mu}}_m) \qquad (33e)$$

$$\hat{\boldsymbol{\chi}}_m = \left[\hat{\boldsymbol{\mu}}_m + \sum_{\ell=1}^{L} a_{m,\ell}\,\mathbf{d}_\ell\right] \bmod \Lambda_{\mathrm{C}} \qquad (33f)$$

$$\hat{\mathbf{s}}_m = Q_{\Lambda_{\mathrm{C}}}(\tilde{\mathbf{y}}_m - \hat{\boldsymbol{\chi}}_m) + \hat{\boldsymbol{\chi}}_m \qquad (33g)$$

The following lemma captures the rates achievable for directly recovering the linear combinations with coefficient matrix $\mathbf{A}$ via successive computation.

Lemma 13: Consider any choice of rates $R_1, \ldots, R_L$ and parameter $\zeta > 0$ and assume that the transmitters and receiver employ the encoding and decoding operations from (28), (29), and (33). For n and prime p large enough, there exists a generator matrix $\mathbf{G} \in \mathbb{Z}_p^{k_{\mathrm{F}} \times n}$ and corresponding nested lattice codes $\mathcal{L}_1, \ldots, \mathcal{L}_L$ with rates at least $R_1 - \zeta, \ldots, R_L - \zeta$, respectively, such that, for any choice of channel matrix $\mathbf{H} \in \mathbb{R}^{N_{\mathrm{r}} \times L}$, integer matrix $\mathbf{A} \in \mathbb{Z}^{L \times L}$, and admissible mapping $\mathcal{I}$, the receiver can recover the linear combinations $\mathbf{u}_1, \ldots, \mathbf{u}_L$ with probability of error at most ζ so long as

$$R_\ell < \frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1})}\right) \quad \forall (m, \ell) \in \mathcal{I} \qquad (34)$$

for some choice of equalization vectors $\mathbf{b}_m \in \mathbb{R}^{N_{\mathrm{r}}}$ and $\mathbf{c}_m \in \mathbb{R}^{m-1}$. □

Proof: The codebook generation process is nearly identical to that in the proof of Lemma 10, except that we set $\epsilon = \zeta/(2L)$ (and keep the choice $\sigma^2_{\text{eff},\ell} = P_\ell 2^{-2R_\ell}$). The encoding process is also identical, and so the power constraint is met as in the proof of Lemma 10. Finally, the decoding step (29) to recover $\mathbf{u}_1$ is identical as well and, as argued in the proof of Lemma 10, $\hat{\mathbf{u}}_1 = \mathbf{u}_1$ with probability of error at most ϵ. We condition on the event that $\hat{\mathbf{u}}_1 = \mathbf{u}_1$ for the remainder of the proof.

To show that $\hat{\mathbf{s}}_1 = \mathbf{a}_1^{\mathsf{T}}\mathbf{X}$, we need to argue that the effective noise $(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}$ is contained in the Voronoi region of the coarsest lattice $\mathcal{V}_{\mathrm{C}}$ so that we can invoke Lemma 11. If at least one rate is non-zero, then we have that $\sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m) < P_{\max}$. From Theorem 8(c), it follows that the effective noise leaves $\mathcal{V}_{\mathrm{C}}$ with probability at most ϵ. Therefore, the decoder can recover $\mathbf{a}_1^{\mathsf{T}}\mathbf{X}$ with probability at most 2ϵ.

We now proceed to argue by induction. Assume that decoding has been successful for steps 1 through $m - 1$ with total probability of error at most $2(m - 1)\epsilon$. We would like to argue that step m is successful with probability of error at most $2m\epsilon$. Define the successive effective noise as

$$\begin{aligned}\mathbf{z}^{\mathsf{T}}_{\text{succ},m} &\triangleq (\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{c}_m^{\mathsf{T}}\hat{\mathbf{S}}_{m-1} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= (\mathbf{b}_m^{\mathsf{T}}\mathbf{H} + \mathbf{c}_m^{\mathsf{T}}\mathbf{A}_{m-1} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}\end{aligned}$$

where the last step uses the correct decoding assumption, $\hat{\mathbf{S}}_{m-1} = [\mathbf{s}_1 \ \cdots \ \mathbf{s}_{m-1}]^{\mathsf{T}} = \mathbf{A}_{m-1}\mathbf{X}$.

The following equations show that the argument inside the quantizer in (33d) can be written as $\boldsymbol{\nu}_m$ plus noise:

$$\begin{aligned}&\left[\tilde{\mathbf{y}}_m + \sum_{i=1}^{m-1} \bar{l}_{m,i}\hat{\boldsymbol{\mu}}_i - \sum_{\ell=1}^{L} a_{m,\ell}\,\mathbf{d}_\ell\right] \bmod \Lambda_{\mathrm{C}} \\ &\overset{(i)}{=} \left[\tilde{\mathbf{y}}_m + \sum_{i=1}^{m-1} \bar{l}_{m,i}\boldsymbol{\mu}_i - \sum_{\ell=1}^{L} a_{m,\ell}\,\mathbf{d}_\ell\right] \bmod \Lambda_{\mathrm{C}} \\ &= \left[\sum_{\ell=1}^{L} a_{m,\ell}(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{i=1}^{m-1} \bar{l}_{m,i}\boldsymbol{\mu}_i + \mathbf{z}_{\text{succ},m}\right] \bmod \Lambda_{\mathrm{C}}\end{aligned}$$

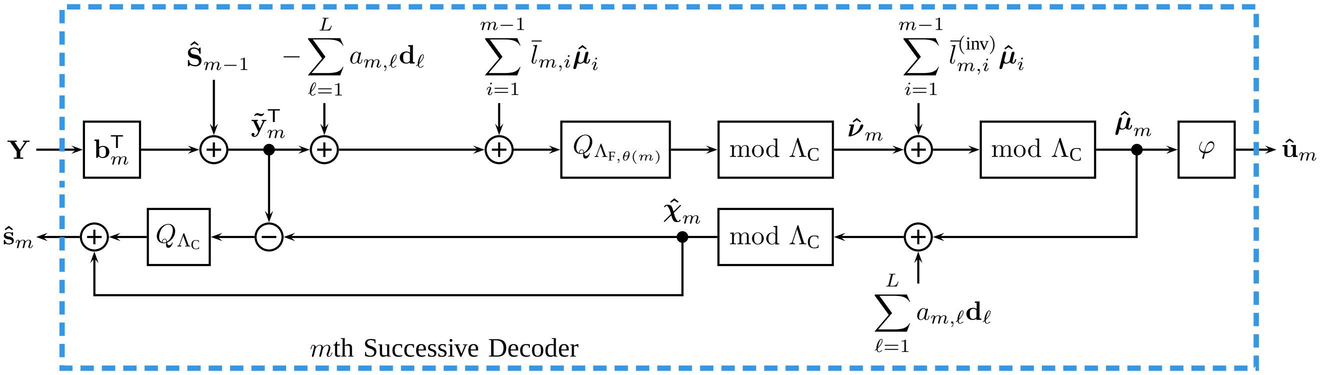
$\mathbf{Y}$
$\mathbf{b}_m^{\mathsf{T}}$
$\hat{\mathbf{S}}_{m-1}$
$\tilde{\mathbf{y}}_m^{\mathsf{T}}$
$-\sum_{\ell=1}^{L} a_{m,\ell}\mathbf{d}_\ell$
$\sum_{i=1}^{m-1} \bar{l}_{m,i}\hat{\boldsymbol{\mu}}_i$
$Q_{\Lambda_{\mathrm{F},\theta(m)}}$
$\bmod \Lambda_{\mathrm{C}}$
$\hat{\boldsymbol{\nu}}_m$
$\sum_{i=1}^{m-1} \bar{l}_{m,i}^{(\mathrm{inv})}\hat{\boldsymbol{\mu}}_i$
$\bmod \Lambda_{\mathrm{C}}$
$\hat{\boldsymbol{\mu}}_m$
φ
$\hat{\mathbf{u}}_m$
$\hat{\mathbf{s}}_m$
$Q_{\Lambda_{\mathrm{C}}}$
$\hat{\boldsymbol{\chi}}_m$
$\bmod \Lambda_{\mathrm{C}}$
$\sum_{\ell=1}^{L} a_{m,\ell}\mathbf{d}_\ell$
mth Successive Decoder

Fig. 10. Block diagram for the mth decoder for successive computation.

$$\overset{(ii)}{=} \left[\boldsymbol{\mu}_m + \sum_{i=1}^{m-1} \bar{l}_{m,i}\boldsymbol{\mu}_i + \mathbf{z}_{\text{succ},m}\right] \bmod \Lambda_{\mathrm{C}}$$
$$\overset{(iii)}{=} \left[\boldsymbol{\nu}_m + \mathbf{z}_{\text{succ},m}\right] \bmod \Lambda_{\mathrm{C}}$$

where (i) uses the assumption that prior decoding steps are correct, (ii) uses the distributive law, and (iii) follows from (31). Combining this with the nested quantization property from (19), we find that

$$\hat{\boldsymbol{\nu}}_m = \left[Q_{\Lambda_{\mathrm{F},\vartheta(m)}}(\boldsymbol{\nu}_m + \mathbf{z}_{\text{succ},m})\right] \bmod \Lambda_{\mathrm{C}} \ .$$

Following a similar labeling argument as in the proof of Lemma 10, it can be shown that $\boldsymbol{\nu}_m \in \Lambda_{\mathrm{F},\vartheta(m)}$. Thus, by Theorem 8(c), we have that $\mathbb{P}(\hat{\boldsymbol{\nu}}_m \neq \boldsymbol{\nu}_m) < \epsilon$ if

$$\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1}) < \sigma^2_{\text{eff},\vartheta(m)} \tag{35}$$

where $\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1})$ is defined in (12). Notice that, if $\hat{\boldsymbol{\nu}}_m = \boldsymbol{\nu}_m$, then $\hat{\boldsymbol{\mu}}_m = \boldsymbol{\mu}_m$ by (32), $\hat{\mathbf{u}}_m = \mathbf{u}_m$ by the linear labeling argument in the proof of Lemma 10. If at least one rate is non-zero, then $\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1}) < P_{\text{max}}$ and we can apply Theorem 8 to establish that the effective noise $\mathbf{z}_{\text{succ},m}$ leaves $\mathcal{V}_{\mathrm{C}}$ with probability at most ϵ. Assuming this is the case, we can invoke Lemma 11 to show that $\hat{\mathbf{s}}_m^{\mathsf{T}} = \mathbf{a}_m^{\mathsf{T}}\mathbf{X}$. To complete the induction step, we apply the union bound to show that $\mathbb{P}\big(\cup_{i=1}^{m} \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\big) < 2m\epsilon$. After L decoding steps, we have probability of error at most $2L\epsilon = \zeta$ as desired. Note that the condition in (35) is equivalent to requiring that $R_\ell < \frac{1}{2}\log^+\big(P_\ell/\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_m, \mathbf{c}_m | \mathbf{A}_{m-1})\big)$ for all $(m,\ell) \in \mathcal{I}$. Finally, it is shown in Appendix H that it suffices to use fixed dither vectors. ■

We are now ready to prove Theorem 2.

Proof of Theorem 2: Choose rates $R_1, \ldots, R_L$ as well as nested lattice codebooks via Lemma 13. The transmitters use the encoding strategy from Lemma 13 (which does not depend on the channel matrix $\mathbf{H}$ nor the integer matrix $\mathbf{A}$). For a given channel matrix $\mathbf{H}$, assume the receiver wants linear combinations with integer coefficient matrix $\mathbf{A}$. This implies that there exists an integer matrix $\tilde{\mathbf{A}}$ satisfying $\operatorname{rowspan}(\mathbf{A}) \subseteq \operatorname{rowspan}(\tilde{\mathbf{A}})$ and admissible mapping $\mathcal{I} \in \mathcal{M}(\tilde{\mathbf{A}})$ such that $(R_1, \ldots, R_L) \in \mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I})$. Therefore, the receiver can use the decoding strategy from Lemma 13 with integer matrix $\tilde{\mathbf{A}}$, admissible mapping $\mathcal{I}$, and optimal equalization vectors $\mathbf{b}_{\text{opt},m}$ and $\mathbf{c}_{\text{opt},m}$ chosen via Lemma 6. Recall from (14) that $\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m | \mathbf{A}_{m-1}) \triangleq \sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m, \mathbf{b}_{\text{opt},m}, \mathbf{c}_{\text{opt},m} | \mathbf{A}_{m-1})$ so (34) matches the rate constraints from $\mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I})$. ■

VII. Conclusions

Although compute-and-forward was originally proposed as a relaying strategy [1], recent efforts have demonstrated that it can be useful in the context of interference alignment [9]–[11] and low-complexity MIMO transceivers [3]–[8]. The aim of this paper was to create a unified framework that captures techniques (e.g., unequal power allocation, successive decoding) that are useful in these settings. Follow-up efforts have employed this expanded framework to develop a notion of uplink-downlink duality for integer-forcing [17] as well as investigate compute-and-forward for discrete memoryless networks [18].

As mentioned earlier, the results of this paper are directly applicable to complex-valued channels by working with a real-valued decomposition of the channel, which corresponds to approximating the complex channel gains with Gaussian integers. However, recent efforts have demonstrated the advantages of working directly over the complex field, e.g., by building lattices from Eisenstein integers [27]. An interesting direction for future work is to generalize the results of Huang *et al.* [27] to create an expanded compute-and-forward framework for Eisenstein integers. More generally, it is of interest to generalize the algebraic framework of Feng *et al.* [14] to permit unequal powers.

Another issue for future study is the development of optimization techniques for use within the expanded compute-and-forward framework. It is well-established (see, for instance, [3], [14]) that the LLL basis reduction algorithm [88] and its variants are an attractive low-complexity solution for the problem of identifying good integer coefficient vectors within the original compute-and-forward framework. These techniques are a natural fit for the parallel computation strategy in Theorem 1. For the successive computation strategy in Theorem 2, it has been shown [7, Theorem 3] that the problem of minimizing the effective noise variances can be linked to Korkin-Zolotarev basis reduction [15]. This can in turn be approximated by multiple applications of the LLL algorithm. More broadly, applications such as integer-forcing interference alignment [10] require simultaneous optimization of beamforming, equalization, and integer coefficient vectors. New heuristics and approximation algorithms are needed to characterize the performance of this strategy versus existing alignment strategies. See [89] for a recent approach that builds on uplink-downlink duality.

Acknowledgments

The authors gratefully acknowledge valuable discussions with U. Erez, C. Feng, M. Gastpar, U. Niesen, O. Ordentlich, and R. Zamir that helped shape this work. In particular, the authors would like to thank C. Feng for his help in adapting the linear labeling results of [14] to form Section IV-C.

Appendix A
Proof of Lemma 2

Rewriting (9), we have that

$$\begin{aligned}\sigma^2_{\text{para}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_m) &= \mathbf{b}_m^{\mathsf{T}}\mathbf{b}_m + \big(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}}\big)\mathbf{P}\big(\mathbf{H}^{\mathsf{T}}\mathbf{b}_m - \mathbf{a}_m\big)\\ &= \mathbf{b}_m^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)\mathbf{b}_m - \mathbf{a}_m^{\mathsf{T}}\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{b}_m\\ &\quad - \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{P}\mathbf{a}_m + \mathbf{a}_m^{\mathsf{T}}\mathbf{P}\mathbf{a}_m \ , \qquad (36)\end{aligned}$$

which has a positive definite Hessian with respect to $\mathbf{b}_m$. Therefore, the minimizing value $\mathbf{b}_{\text{opt},m}$ is found by setting the derivative equal to zero:

$$\begin{aligned}2\mathbf{b}_{\text{opt},m}^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big) - 2\mathbf{a}_m^{\mathsf{T}}\mathbf{P}\mathbf{H}^{\mathsf{T}} &= \mathbf{0}\\ \mathbf{b}_{\text{opt},m}^{\mathsf{T}} &= \mathbf{a}_m^{\mathsf{T}}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1} \ .\end{aligned}$$

Plugging this back into (36), we get that

$$\begin{aligned}&\min_{\mathbf{b}_m\in\mathbb{R}^{N_\text{r}}} \sigma^2_{\text{para}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_m)\\ &= \mathbf{a}_m^{\mathsf{T}}\big(\mathbf{P} - \mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)\mathbf{H}\mathbf{P}\big)\mathbf{a}_m\end{aligned}$$

from which the result follows by applying (8) and (7). ■

Appendix B
Proof of Lemma 3

Assume, for the sake of the contradiction, that $\tilde{a}^2_{m,\ell} > \lambda_{\text{max}}\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)$ for some m and that there is a rate tuple in $\mathcal{R}_{\text{para}}(\mathbf{H},\tilde{\mathbf{A}})$ for which $R_\ell > 0$. Starting from the rate expression in Theorem 1, $R_\ell > 0$ implies that

$$\begin{aligned}P_\ell &> \sigma^2_{\text{para}}(\mathbf{H},\tilde{\mathbf{a}}_m)\\ &= \tilde{\mathbf{a}}_m^{\mathsf{T}}\big(\mathbf{P}^{-1}+\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)^{-1}\tilde{\mathbf{a}}_m\\ &= \tilde{\mathbf{a}}_m^{\mathsf{T}}\mathbf{P}\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)^{-1}\tilde{\mathbf{a}}_m\\ &\overset{\text{(i)}}{\geq} \tilde{\mathbf{a}}_m^{\mathsf{T}}\mathbf{P}\tilde{\mathbf{a}}_m\lambda_{\text{min}}\Big(\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)^{-1}\Big)\\ &= \frac{\tilde{\mathbf{a}}_m^{\mathsf{T}}\mathbf{P}\tilde{\mathbf{a}}_m}{\lambda_{\text{max}}\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)}\\ &\geq \frac{P_\ell\tilde{a}^2_{m,\ell}}{\lambda_{\text{max}}\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)}\\ &\implies \lambda_{\text{max}}\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big) \geq \tilde{a}^2_{m,\ell}\end{aligned}$$

where in (i) $\lambda_{\text{min}}\Big(\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)^{-1}\Big)$ refers to the minimum eigenvalue of $\big(\mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{H}\big)^{-1}$ and the inequality is due to the Min-Max Theorem [90, Theorem 4.2.2] for symmetric matrices. Thus, a contradiction has been reached, which establishes the desired bound on $\tilde{a}^2_{m,\ell}$. ■

Appendix C
Proof of Lemma 6

From (12), it follows that

$$\begin{aligned}&\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_m,\mathbf{c}_m|\mathbf{A}_{m-1})\\ &\ = \mathbf{b}_m^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)\mathbf{b}_m \ - \ 2\mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{P}\big(\mathbf{a}_m - \mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m\big)\\ &\quad + \ \big(\mathbf{a}_m^{\mathsf{T}} - \mathbf{c}_m^{\mathsf{T}}\mathbf{A}_{m-1}\big)\mathbf{P}\big(\mathbf{a}_m - \mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m\big) \ . \qquad (37)\end{aligned}$$

This expression has a positive definite Hessian with respect to the vector $[\mathbf{b}_m^{\mathsf{T}} \ \mathbf{c}_m^{\mathsf{T}}]$. Therefore, the optimizing vector can be found by setting the derivative equal to zero. We start by solving for $\mathbf{b}_{\text{opt},m}$ in terms of $\mathbf{c}_{\text{opt},m}$.

Taking the derivative of (37) with respect to $\mathbf{b}_m$ and setting it equal to zero, we obtain

$$2\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)\mathbf{b}_m - 2\mathbf{H}\mathbf{P}\big(\mathbf{a}_m - \mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m\big) = \mathbf{0} \ .$$

It follows that

$$\mathbf{b}_{\text{opt},m}^{\mathsf{T}} = \big(\mathbf{a}_m^{\mathsf{T}} - \mathbf{c}_m^{\mathsf{T}}\mathbf{A}_{m-1}\big)\mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1} \ .$$

Plugging back into (37) and canceling terms, we get

$$\begin{aligned}&\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_{\text{opt},m},\mathbf{c}_m|\mathbf{A}_{m-1})\\ &= \big(\mathbf{a}_m^{\mathsf{T}} - \mathbf{c}_m^{\mathsf{T}}\mathbf{A}_{m-1}\big)\big(\mathbf{P} - \mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{I}+\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}\big)^{-1}\mathbf{H}\mathbf{P}\big)\big(\mathbf{a}_m - \mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m\big)\\ &= \big(\mathbf{a}_m^{\mathsf{T}} - \mathbf{c}_m^{\mathsf{T}}\mathbf{A}_{m-1}\big)\mathbf{F}^{\mathsf{T}}\mathbf{F}\big(\mathbf{a}_m - \mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m\big) \ . \qquad (38)\end{aligned}$$

where the last step uses (8) and (7). Next, we take the derivative with respect to $\mathbf{c}_m$ and set it equal to zero,

$$2\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\mathbf{c}_m - 2\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{a}_m = \mathbf{0}$$

from which it follows that

$$\mathbf{c}_{\text{opt},m}^{\mathsf{T}} = \mathbf{a}_m^{\mathsf{T}}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big(\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}\big)^{-1} \ .$$

Finally, we plug back into (38) and cancel terms to obtain

$$\begin{aligned}&\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_{\text{opt},m},\mathbf{c}_{\text{opt},m}|\mathbf{A}_{m-1})\\ &= \mathbf{a}_m^{\mathsf{T}}\big(\mathbf{I} - \mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big(\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big)^{-1}\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\big)\mathbf{F}\mathbf{a}_m \ .\end{aligned}$$

Note that $\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big(\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big)^{-1}\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}$ is the projection matrix for the subspace spanned by $\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}$ and $\mathbf{N}_{m-1} = \mathbf{I} - \mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big(\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{A}_{m-1}^{\mathsf{T}}\big)^{-1}\mathbf{A}_{m-1}\mathbf{F}^{\mathsf{T}}\big)\mathbf{F}$ is the projection matrix for the corresponding nullspace. Since projection matrices are idempotent (i.e, $\mathbf{N}^2_{m-1} = \mathbf{N}_{m-1}$) and $\mathbf{N}_{m-1}$ is symmetric, it follows that

$$\begin{aligned}\sigma^2_{\text{succ}}(\mathbf{H},\mathbf{a}_m,\mathbf{b}_{\text{opt},m},\mathbf{c}_{\text{opt},m}|\mathbf{A}_{m-1}) &= \mathbf{a}_m^{\mathsf{T}}\mathbf{F}^{\mathsf{T}}\mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m\\ &= \mathbf{a}_m^{\mathsf{T}}\mathbf{F}^{\mathsf{T}}\mathbf{N}_{m-1}^{\mathsf{T}}\mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m\\ &= \big\|\mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m\big\|^2 \ .\end{aligned}$$

■

Appendix D
Proof of Theorem 3

The inclusion $\mathcal{R}^{\text{(prim)}}_{\text{comp}}(\mathbf{H},\mathbf{A}) \subseteq \mathcal{R}^{\text{(succ)}}_{\text{comp}}(\mathbf{H},\mathbf{A})$ follows directly from the fact that the union in the former computation rate region is taken over a subset of the matrices $\tilde{\mathbf{A}}$ used in the union in the latter. We now turn to argue that $\mathcal{R}^{\text{(prim)}}_{\text{comp}}(\mathbf{H},\mathbf{A}) \supseteq \mathcal{R}^{\text{(succ)}}_{\text{comp}}(\mathbf{H},\mathbf{A})$. In particular, we will show that for any integer matrix $\tilde{\mathbf{A}}$, there exists a primitive basis matrix $\tilde{\mathbf{A}}_{\text{prim}}$ with the same rowspan such that $\mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}, \mathcal{I}) \subseteq \mathcal{R}_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{A}}_{\text{prim}}, \mathcal{I})$ for any admissible mapping $\mathcal{I}$ with respect to $\tilde{\mathbf{A}}$. Without loss of generality, we assume that $\tilde{\mathbf{A}}$ is of the form

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}_M \\ \mathbf{0}_{(L-M)\times L} \end{bmatrix} \tag{39}$$

where the submatrix $\tilde{\mathbf{A}}_M$ has $M \leq L$ rows and is full rank.

Lemma 14: For any integer matrix $\tilde{\mathbf{A}}$ of the form (39), there exists a rank M primitive basis matrix $\tilde{\mathbf{A}}_{\text{prim}} = \begin{bmatrix} \tilde{\mathbf{A}}_{\text{prim},M} \\ \mathbf{0}_{(L-M)\times L} \end{bmatrix}$ such that $\tilde{\mathbf{A}}_M = \mathbf{T}\tilde{\mathbf{A}}_{\text{prim},M}$ where $\mathbf{T}$ is an $M \times M$ lower triangular integer matrix with strictly positive diagonal entries. □

The proof follows directly from [77, Corollary 1.24].

Let $\tilde{\mathbf{A}}_{\text{prim}}$ be a primitive basis matrix chosen using the lemma above. By assumption, there exists a lower unitriangular matrix $\mathbf{L}$ that shows that $\mathcal{I}$ is admissible for $\tilde{\mathbf{A}}$. Specifically, for each index pair $(i,j) \notin \mathcal{I}$, we have that $\mathbf{l}_i^\mathsf{T}\tilde{\mathbf{a}}_j = 0$ where $\mathbf{l}_i^\mathsf{T}$ is the ith row of $\mathbf{L}$. We would like to show that $\mathcal{I}$ is admissible for $\tilde{\mathbf{A}}_{\text{prim}}$ as well. Define

$$\tilde{\mathbf{L}} = \mathbf{L}\begin{bmatrix} \mathbf{T} & \mathbf{0}_{M\times(L-M)} \\ \mathbf{0}_{(L-M)\times M} & \mathbf{I}_L \end{bmatrix}$$

and note that $\tilde{\mathbf{L}}$ is lower triangular and $\tilde{\mathbf{l}}_i^\mathsf{T}\tilde{\mathbf{a}}_{\text{prim},j} = 0$ where $\tilde{\mathbf{l}}_i^\mathsf{T}$ is the ith row of $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{a}}_{\text{prim},j}$ is the jth column of $\tilde{\mathbf{A}}_{\text{prim}}$. Finally, we renormalize the rows to obtain a unitriangular matrix $\mathbf{L}_{\text{prim}}$ whose ith row is equal to $\mathbf{l}_{\text{prim},i}^\mathsf{T} = (\tilde{l}_{i,i})^{-1}\tilde{\mathbf{l}}_i^\mathsf{T}$. We still have that $\mathbf{l}_{\text{prim},i}^\mathsf{T}\tilde{\mathbf{a}}_{\text{prim},j} = 0$ so $\mathcal{I}$ is admissible for $\tilde{\mathbf{A}}_{\text{prim}}$.

To complete the proof, we need to argue that the effective noise variances can only decrease by using $\tilde{\mathbf{A}}_{\text{prim}}$ instead of $\tilde{\mathbf{A}}$. Let $t_{i,j}$ denote the (i,j)th entry of $\mathbf{T}$ and $\mathbf{N}_{\text{prim},m-1}$ the nullspace projection matrix (13) for $\tilde{\mathbf{A}}_{\text{prim}}$. Starting from (14), we have that

$$\begin{aligned}
&\sigma^2_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{a}}_m | \tilde{\mathbf{A}}_{m-1}) \\
&= \min_{\mathbf{b}_m} \|\mathbf{b}_m\|^2 + \left\|\left(\mathbf{b}_m^\mathsf{T}\mathbf{H} - \mathbf{c}_{\text{opt},m}^\mathsf{T}\tilde{\mathbf{A}}_{m-1} - \tilde{\mathbf{a}}_m^\mathsf{T}\right)\mathbf{P}^{1/2}\right\|^2 \\
&\overset{(i)}{=} \min_{\mathbf{b}_m} \|\mathbf{b}_m\|^2 + \left\|\left(\mathbf{b}_m^\mathsf{T}\mathbf{H} - \tilde{\mathbf{c}}^\mathsf{T}\tilde{\mathbf{A}}_{\text{prim},m-1} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T}\right)\mathbf{P}^{1/2}\right\|^2 \\
&\geq \min_{\mathbf{b}_m,\mathbf{c}_m} \|\mathbf{b}_m\|^2 + \left\|\left(\mathbf{b}_m^\mathsf{T}\mathbf{H} - \mathbf{c}_m^\mathsf{T}\tilde{\mathbf{A}}_{\text{prim},m-1} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T}\right)\mathbf{P}^{1/2}\right\|^2 \\
&= \sigma^2_{\text{succ}}(\mathbf{H}, t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m} | \tilde{\mathbf{A}}_{m-1}) \\
&= \left\|\mathbf{N}_{\text{prim},m-1}\mathbf{F}t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}\right\|^2 \\
&\overset{(ii)}{\geq} \left\|\mathbf{N}_{\text{prim},m-1}\mathbf{F}\tilde{\mathbf{a}}_{\text{prim},m}\right\|^2 \\
&= \sigma^2_{\text{succ}}(\mathbf{H}, \tilde{\mathbf{a}}_{\text{prim},m} | \tilde{\mathbf{A}}_{\text{prim},m-1})
\end{aligned}$$

where (i) relies on the fact that $\mathbf{c}_{\text{opt},m}^\mathsf{T}\tilde{\mathbf{A}}_{m-1} - \tilde{\mathbf{a}}_m^\mathsf{T} = \tilde{\mathbf{c}}^\mathsf{T}\tilde{\mathbf{A}}_{\text{prim},m-1} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T}$ for some choice of $\tilde{\mathbf{c}}$ which is shown below and (ii) uses the fact that $t_{m,m} \geq 1$ from Lemma 14.

Let the jth entry of $\tilde{\mathbf{c}}$ be $\tilde{c}_j = \sum_{i=j}^{m-1} t_{i,j}c_{\text{opt},m,i} - t_{m,j}$ where $c_{\text{opt},m,i}$ is the ith entry of $\mathbf{c}_{\text{opt},m}$. It follows that

$$\begin{aligned}
&\tilde{\mathbf{c}}^\mathsf{T}\tilde{\mathbf{A}}_{\text{prim},m-1} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T} \\
&= \sum_{j=1}^{m-1} \tilde{c}_j\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T} \\
&= \sum_{j=1}^{m-1}\left(\sum_{i=j}^{m-1} t_{i,j}c_{\text{opt},m,i} - t_{m,j}\right)\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} - t_{m,m}\tilde{\mathbf{a}}_{\text{prim},m}^\mathsf{T} \\
&= \sum_{j=1}^{m-1}\left(\sum_{i=j}^{m-1} t_{i,j}c_{\text{opt},m,i}\right)\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} - \sum_{j=1}^{m} t_{m,j}\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} \\
&= \sum_{i=1}^{m-1} c_{\text{opt},m,i}\sum_{j=1}^{i} t_{i,j}\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} - \sum_{j=1}^{m} t_{m,j}\tilde{\mathbf{a}}_{\text{prim},j}^\mathsf{T} \\
&\overset{(i)}{=} \sum_{i=1}^{m-1} c_{\text{opt},m,i}\tilde{\mathbf{a}}_i^\mathsf{T} - \tilde{\mathbf{a}}_m^\mathsf{T} \\
&= \mathbf{c}_{\text{opt},m}^\mathsf{T}\tilde{\mathbf{A}}_{m-1} - \tilde{\mathbf{a}}_m^\mathsf{T}
\end{aligned}$$

where (i) uses the fact that $\tilde{\mathbf{A}}_M = \mathbf{T}\tilde{\mathbf{A}}_{\text{prim},M}$ and that $\mathbf{T}$ is lower triangular. ■

Appendix E
Proof of Theorem 4

Our proof follows along the same lines as the proof for the equal power setting [9, Theorem 3]. The following definition and theorem specialize basic results from the geometry of numbers [91] to the Euclidean case.

Definition 12 (Successive Minima): Let Λ be a full-rank lattice in $\mathbb{R}^L$. For $m = 1, \ldots, L$, the mth *successive minimum* λ_m of Λ corresponds to the radius of the smallest Euclidean ball centered at the origin that captures m linearly independent lattice points,

$$\lambda_m \triangleq \inf\left\{r > 0 : \dim\operatorname{span}\big(\mathcal{B}(\mathbf{0}, r) \cap \Lambda\big) = m\right\} .$$

The following lemma is a special case of Minkowski's Second Theorem [91, p.156].

Lemma 15 ([15, Theorem 1.5]): Let $\mathbf{F} \in \mathbb{R}^{L\times L}$ be a full-rank matrix and let $\Lambda = \mathbf{F}\mathbb{Z}^L$ be the resulting full-rank lattice. The product of the successive minima is upper bounded as follows:

$$\prod_{m=1}^{L} \lambda_m^2 \leq L^L\big|\det(\mathbf{F})\big|^2$$

Now, let $\Lambda_{\text{channel}} = \mathbf{F}\mathbb{Z}^L$ be the full-rank lattice generated by $\mathbf{F}$ from (7). Notice that the lengths of a dominant solution $\mathbf{a}_1^*, \ldots, \mathbf{a}_L^*$ correspond exactly to the successive minima of this lattice, $\lambda_m = \|\mathbf{F}\mathbf{a}_m^*\|$. Furthermore, recall that the determinant of a lattice is defined as the absolute value of the determinant of any basis for the lattice [60, Definition 2.1.2] so that $\det(\Lambda_{\text{channel}}) = |\det(\mathbf{F})|$.

We now lower bound the sum of the rates from the theorem statement.

$$\begin{aligned}
&\sum_{\ell=1}^{L}\frac{1}{2}\log^+\left(\frac{P_\ell}{\sigma^2_{\text{para}}\big(\mathbf{H}, \mathbf{a}^*_{\pi(\ell)}\big)}\right) \\
&\geq \sum_{\ell=1}^{L}\frac{1}{2}\log\left(\frac{P_\ell}{\sigma^2_{\text{para}}\big(\mathbf{H}, \mathbf{a}^*_{\pi(\ell)}\big)}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\log\left(\prod_{\ell=1}^{L} P_\ell\right) - \frac{1}{2}\log\left(\prod_{\ell=1}^{L} \sigma^2_{\text{para}}(\mathbf{H}, \mathbf{a}^*_{\pi(\ell)})\right) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\left(\prod_{m=1}^{L} \|\mathbf{F}\mathbf{a}^*_m\|^2\right) \\
&\overset{(i)}{\geq} \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\left(L^L|\det(\mathbf{F})|^2\right) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\det(\mathbf{F}^\mathsf{T}\mathbf{F}) - \frac{L}{2}\log(L) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\det\Big(\big(\mathbf{P}^{-1} + \mathbf{H}^\mathsf{T}\mathbf{H}\big)^{-1}\Big) - \frac{L}{2}\log(L) \\
&= \frac{1}{2}\log\det\big(\mathbf{I} + \mathbf{P}\mathbf{H}^\mathsf{T}\mathbf{H}\big) - \frac{L}{2}\log(L) \\
&\overset{(ii)}{=} \frac{1}{2}\log\det\big(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T}\big) - \frac{L}{2}\log(L)
\end{aligned}
$$

where (i) is due to Minkowski's Second Theorem from above and (ii) is due to Sylvester's Determinant Identity [92]: for any square matrices $\mathbf{M}_1$ and $\mathbf{M}_2$,

$$
\det\big(\mathbf{I} + \mathbf{M}_1\mathbf{M}_2\big) = \det\big(\mathbf{I} + \mathbf{M}_2\mathbf{M}_1\big) \ . \tag{40}
$$

■

Appendix F
Proof of Lemma 8

First, note that the vectors $\mathbf{F}\mathbf{a}_1, \ldots, \mathbf{F}\mathbf{a}_L$ form a basis of $\mathbb{R}^L$ since $\mathbf{F}$ and $\mathbf{A}$ have rank L. Denote the Gram-Schmidt orthogonalization of this basis by $\mathbf{g}^*_1, \ldots, \mathbf{g}^*_L$ where

$$
\mathbf{g}^*_m = \mathbf{N}_{m-1}\mathbf{F}\mathbf{a}_m
$$

and $\mathbf{N}_{m-1}$ refers to the projection matrix for the nullspace of $\mathbf{F}\mathbf{A}^\mathsf{T}_{m-1}$ defined in (13). Define $\mathbf{G}^* = [\mathbf{g}^*_1 \ \cdots \ \mathbf{g}^*_L]^\mathsf{T}$. It can be shown [77, Theorem 3.4] that $\det(\mathbf{G}^*) = \det(\mathbf{F}\mathbf{A}^\mathsf{T})$.

Since the Gram-Schmidt vectors are orthogonal, we have that

$$
\det\big(\mathbf{G}^*(\mathbf{G}^*)^\mathsf{T}\big) = \prod_{m=1}^{L}\|\mathbf{g}^*_m\|^2 = \prod_{m=1}^{L}\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m|\mathbf{A}_{m-1}) \ .
$$

It follows that

$$
\begin{aligned}
&\sum_{\ell=1}^{L}\frac{1}{2}\log\left(\frac{P_\ell}{\sigma^2_{\text{succ}}\big(\mathbf{H}, \mathbf{a}_{\pi(\ell)}|\mathbf{A}_{\pi(\ell)-1}\big)}\right) \\
&= \frac{1}{2}\log\left(\prod_{\ell=1}^{L}P_\ell\right) - \frac{1}{2}\log\left(\prod_{m=1}^{L}\sigma^2_{\text{succ}}(\mathbf{H}, \mathbf{a}_m|\mathbf{A}_{m-1})\right) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\det\big(\mathbf{G}^*(\mathbf{G}^*)^\mathsf{T}\big) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\det(\mathbf{A}\mathbf{F}^\mathsf{T}\mathbf{F}\mathbf{A}^\mathsf{T}) \\
&= \frac{1}{2}\log\det(\mathbf{P}) - \frac{1}{2}\log\det\Big(\big(\mathbf{P}^{-1} + \mathbf{H}^\mathsf{T}\mathbf{H}\big)^{-1}\Big) - \log\det(\mathbf{A}) \\
&\overset{(i)}{=} \frac{1}{2}\log\det\big(\mathbf{I} + \mathbf{P}\mathbf{H}^\mathsf{T}\mathbf{H}\big) \\
&\overset{(ii)}{=} \frac{1}{2}\log\det\big(\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T}\big)
\end{aligned}
$$

where (i) uses the fact that $\mathbf{A}$ is unimodular so $\det(\mathbf{A}) = 1$ and (ii) uses Sylvester's Determinant Identity from (40). ■

Appendix G
Proof of Theorem 9

The following lemma will be useful for the proof.

Lemma 16: For $\ell = 1, \ldots, L$, let $\boldsymbol{\lambda}_\ell \in \Lambda_{\text{F}}$ be lattice codewords with corresponding linear codewords $\bar{\phi}(\boldsymbol{\lambda}_\ell)$. Then, for any integer combination $\boldsymbol{\mu} = \sum_{\ell=1}^{L} a_\ell \boldsymbol{\lambda}_\ell$ where $a_\ell \in \mathbb{Z}$, the corresponding linear codeword $\bar{\phi}(\boldsymbol{\mu})$ satisfies

$$
\bar{\phi}(\boldsymbol{\mu}) = \bigoplus_{\ell=1}^{L} q_\ell \, \bar{\phi}(\boldsymbol{\lambda}_\ell)
$$

where $q_\ell = [a_\ell] \bmod p$. □

Proof:

$$
\begin{aligned}
\bar{\phi}(\boldsymbol{\mu}) &= \bar{\phi}\bigg(\sum_{\ell=1}^{L} a_\ell\boldsymbol{\lambda}_\ell\bigg) \\
&= \left[\gamma^{-1}p\sum_{\ell=1}^{L} a_\ell\boldsymbol{\lambda}_\ell\right] \bmod p \\
&= [\gamma^{-1}pa_1\boldsymbol{\lambda}_1] \bmod p \oplus \cdots \oplus [\gamma^{-1}pa_L\boldsymbol{\lambda}_L] \bmod p \\
&= [a_1] \bmod p \cdot [\gamma^{-1}p\boldsymbol{\lambda}_1] \bmod p \, \oplus \\
&\quad \cdots \ \oplus [a_L] \bmod p \cdot [\gamma^{-1}p\boldsymbol{\lambda}_L] \bmod p \\
&= q_1 \, \bar{\phi}(\boldsymbol{\lambda}_1) \oplus \cdots \oplus q_L \, \bar{\phi}(\boldsymbol{\lambda}_L)
\end{aligned}
$$

■

We now establish that the proposed labeling φ satisfies Definition 11(a). Recall that $\boldsymbol{\lambda} \in \Lambda_{\text{F},\ell}$ if and only if the corresponding linear codeword $\bar{\phi}(\boldsymbol{\lambda}) \in \mathcal{C}_{\text{F},\ell}$. Let $\mathbf{v} \in \mathbb{Z}_p^{k_{\text{F}}}$ be the unique vector satisfying $\bar{\phi}(\boldsymbol{\lambda}) = \mathbf{G}^\mathsf{T}\mathbf{v}$. If $\bar{\phi}(\boldsymbol{\lambda}) \in \mathcal{C}_{\text{F},\ell}$, then the last $k_{\text{F}} - k_{\text{F},\ell}$ components of $\mathbf{v}$ must be equal to 0, meaning that the last $k_{\text{F}} - k_{\text{F},\ell}$ components of the label $\varphi(\boldsymbol{\lambda})$ are equal to 0. Similarly, $\boldsymbol{\lambda} \in \Lambda_{\text{C},\ell}$ if and only if the last $k_{\text{F}} - k_{\text{C},\ell}$ components of the label $\varphi(\boldsymbol{\lambda})$ are equal to 0.

Now we turn to establish that φ satisfies Definition 11(b). Let $\boldsymbol{\mu} = \sum_{\ell=1}^{L} a_\ell\boldsymbol{\lambda}_\ell$. For each $\boldsymbol{\lambda}_\ell \in \Lambda_{\text{F}}$, let $\mathbf{v}_\ell \in \mathbb{Z}_p^{k_{\text{F}}}$ denote the unique vector that satisfies $\bar{\phi}(\boldsymbol{\lambda}_\ell) = \mathbf{G}^\mathsf{T}\mathbf{v}_\ell$. From Lemma 16, we have that

$$
\begin{aligned}
\bar{\phi}(\boldsymbol{\mu}) &= \bigoplus_{\ell=1}^{L} q_\ell \, \bar{\phi}(\boldsymbol{\lambda}_\ell) \\
&= \bigoplus_{\ell=1}^{L} q_\ell\mathbf{G}^\mathsf{T}\mathbf{v}_\ell \\
&= \mathbf{G}^\mathsf{T}\bigoplus_{\ell=1}^{L} q_\ell\mathbf{v}_\ell
\end{aligned}
$$

where $q_\ell = [a_\ell] \bmod p$. Let $\mathbf{t} \in \mathbb{Z}_p^{k_{\text{F}}}$ be the unique vector satisfying $\bar{\phi}(\boldsymbol{\mu}) = \mathbf{G}^\mathsf{T}\mathbf{t}$. Since $\mathbf{G}$ is full rank, it follows that

$$
\mathbf{t} = \bigoplus_{\ell=1}^{L} q_\ell\mathbf{v}_\ell \ .
$$

Finally, since the labels $\varphi(\boldsymbol{\mu})$ and $\varphi(\boldsymbol{\lambda}_\ell)$ consist of the last k elements of $\mathbf{t}$ and $\mathbf{v}_\ell$, respectively, we have that

$$
\varphi(\boldsymbol{\mu}) = \bigoplus_{\ell=1}^{L} q_\ell \, \varphi(\boldsymbol{\lambda}_\ell)
$$

as desired. ■

Appendix H
Fixed Dithers

Assume that we have established Lemma 10 or 13 using random dither vectors. Specifically, each dither vector $\mathbf{d}_\ell$ is independently generated according to a uniform distribution over $\mathcal{V}_\ell$. We would like to establish that there exist fixed dither vectors that can achieve the same rate region. Select any power constraints, channel matrix, and integer matrix as well as any rate tuple in the achievable region. Let $\mathbf{x}_\ell(\mathbf{w}_\ell, \mathbf{d}_\ell)$ denote the channel input sequence for message vector $\mathbf{w}_\ell \in \mathbb{Z}_p^{k_{\text{F},\ell} - k_{\text{C},\ell}}$ and dither vector $\mathbf{d}_\ell \in \mathcal{V}_{\text{C},\ell}$. For a given realization of $\mathbf{d}_\ell$, the average power is

$$P_{\text{avg},\ell}(\mathbf{d}_\ell) = \frac{1}{p^{k_{\text{F},\ell} - k_{\text{C},\ell}}} \sum_{\mathbf{w}_\ell} \left\| \mathbf{x}_\ell(\mathbf{w}_\ell, \mathbf{d}_\ell) \right\|^2$$

where the average is taken with respect to a uniform distribution over possible message vectors. Let $0 < \gamma < 1$ be a parameter to be specified later. Using the fact that $\mathbb{E}[P_{\text{avg},\ell}(\mathbf{d}_\ell)] \le P_\ell$, it follows from Markov's inequality that

$$\mathbb{P}\left(P_{\text{avg},\ell}(\mathbf{d}_\ell) < \frac{P_\ell}{1 - \gamma^{1/L}} \right) > \gamma^{1/L} \ .$$

Using the independence of the dithers, we thus have that

$$\mathbb{P}\left(\bigcap_{\ell=1}^{L} \left\{ P_{\text{avg},\ell}(\mathbf{d}_\ell) < \frac{P_\ell}{1 - \gamma^{1/L}} \right\} \right) > \gamma \ . \tag{41}$$

For a given realization of $\mathbf{d}_1, \ldots, \mathbf{d}_L$, the average probability of error is

$$\begin{aligned} &p_{\text{error}}(\mathbf{d}_1, \ldots, \mathbf{d}_L) \\ &= \frac{1}{p^{\sum_\ell k_{\text{F},\ell} - k_{\text{C},\ell}}} \sum_{\mathbf{w}_1, \ldots, \mathbf{w}_L} \mathbb{1}(\hat{\mathbf{u}}_\ell \neq \mathbf{u}_\ell \text{ for some } \ell) \end{aligned}$$

where the average is taken with respect to a uniform distribution over possible message vectors. Using the fact that $\mathbb{E}[p_{\text{error}}(\mathbf{d}_1, \ldots, \mathbf{d}_L)] \le \epsilon$, we have from Markov's inequality that

$$\mathbb{P}\left(p_{\text{error}}(\mathbf{d}_1, \ldots, \mathbf{d}_L) < \frac{2\epsilon}{\gamma} \right) > 1 - \frac{\gamma}{2} \ .$$

Combining this with (41), we know that with probability $\gamma/2$, each power is at most $\frac{1}{1-\gamma^{1/L}}$ times larger than its original target P_ℓ and the average error probability is at most $\frac{2\epsilon}{\gamma}$. Therefore, there exist fixed dither vectors that satisfy these relaxed constraints as well. By taking γ to zero, we can get as close as needed to the original target powers P_ℓ. Afterwards, we can make the average probability of error as small to be as desired by choosing ϵ. Finally, since the rate expressions are continuous functions of the powers, we can operate as close as we would like to any rate tuple in the original rate region using fixed dithers.

References

[1] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, pp. 6463–6486, October 2011.

[2] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proceedings of the* IEEE, vol. 99, pp. 438–460, March 2011.

[3] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, vol. 60, pp. 7661–7685, December 2014.

[4] S.-N. Hong and G. Caire, "Reverse compute and forward: A low-complexity architecture for downlink distributed antenna systems," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2012)*, (Cambridge, MA), July 2012.

[5] S. N. Hong and G. Caire, "Compute-and-forward strategies for cooperative distributed antenna systems," *IEEE Transactions on Information Theory*, vol. 59, pp. 5227–5243, September 2013.

[6] A. Sakzad, J. Harshan, and E. Viterbo, "Integer-forcing MIMO linear receivers based on lattice reduction," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 4905–4915, October 2013.

[7] O. Ordentlich, U. Erez, and B. Nazer, "Successive integer-forcing and its sum-rate optimality," in *Proceedings of the 51st Annual Allerton Conference on Communications, Control, and Computing*, (Monticello, IL), October 2013.

[8] O. Ordentlich and U. Erez, "Precoded integer-forcing universally achieves the MIMO capacity to within a constant gap," *IEEE Transactions on Information Theory*, vol. 61, pp. 323–340, January 2015.

[9] O. Ordentlich, U. Erez, and B. Nazer, "The approximate sum capacity of the symmetric K-user Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 60, pp. 3450–3482, June 2014.

[10] V. Ntranos, V. Cadambe, B. Nazer, and G. Caire, "Integer-forcing interference alignment," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2013)*, (Istanbul, Turkey), July 2013.

[11] M. Farag and B. Nazer, "The symmetric ergodic capacity of phase-fading interference channels to within a constant gap: 3 users in the strong and very strong regimes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2013)*, (Istanbul, Turkey), July 2013.

[12] W. Nam, S. Chung, and Y. Lee, "Nested lattice codes for Gaussian relay networks with interference," *IEEE Transactions on Information Theory*, vol. 57, pp. 7733–7745, December 2011.

[13] S. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, pp. 1872–1905, April 2011.

[14] C. Feng, D. Silva, and F. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Transactions on Information Theory*, vol. 59, pp. 7576–7596, November 2013.

[15] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, vol. 671 of The Kluwer International International Series in Engineering and Computer Science. Cambridge, UK: Kluwer Academic Publishers, 2002.

[16] B. Nazer, "Successive compute-and-forward," in *Proceedings of the International Zurich Seminar on Communications (IZS 2012)*, (Zurich, Switzerland), March 2012.

[17] W. He, B. Nazer, and S. Shamai (Shitz), "Uplink-downlink duality for integer-forcing," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2014)*, (Honolulu, HI), July 2014.

[18] S. H. Lim, C. Feng, B. Nazer, and M. Gastpar, "A joint typicality approach to compute-forward," in *Proceedings of 53rd Annual Allerton Conference on Communications, Control and Computing*, (Monticello, IL), October 2015.

[19] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[20] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, February 2003.

[21] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, October 2003.

[22] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Proceedings of the IEEE International Conference on Communications (ICC 2006)*, (Istanbul, Turkey), June 2006.

[23] B. Nazer and M. Gastpar, "Computing over multiple-access channels with connections to wireless network coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2006)*, (Seattle, WA), July 2006.

[24] S. Zhang, S.-C. Liew, and P. Lam, "Hot topic: Physical-layer network coding," in *Proceedings of the 12th Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2006)*, (Los Angeles, CA), September 2006.

[25] B. Nazer and M. Gastpar, "Lattice coding increases multicast rates for Gaussian multiple-access networks," in *45th Annual Allerton Conference on Communications, Control, and Computing*, (Monticello, IL), September 2007.

[26] M. P. Wilson, K. Narayanan, H. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *IEEE Transactions on Information Theory*, vol. 11, pp. 5641–5654, November 2010.

[27] N. E. Tunali, Y.-C. Huang, J. J. Boutros, and K. R. Narayanan, "Lattices over Eisenstein integers for compute-and-forward," *IEEE Transactions on Information Theory*, vol. 61, pp. 5306–5321, October 2015.

[28] J.-C. Belfiore and C. Ling, "The flatness factor in lattice network coding: Design criterion and decoding algorithm," in *Proceedings of the International Zurich Seminar on Communications (IZS 2012)*, (Zurich, Switzerland), March 2012.

[29] Z. Faraji-Dana and P. Mitran, "On non-binary constellations for channel-coded physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 312–319, January 2013.

[30] N. E. Tunali, K. R. Narayanan, and H. D. Pfister, "Spatially-coupled low density lattices based on construction A with applications to compute-and-forward," in *Proceedings of the IEEE Information Theory Workshop (ITW 2013)*, (Seville, Spain), September 2013.

[31] Y.-C. Huang, K. R. Narayanan, and N. E. Tunali, "Multistage compute-and-forward with multilevel lattice codes based on product constructions," *IEEE Transactions on Information Theory*, Submitted January 2014. http://arxiv.org/abs/1401.2228.

[32] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Transactions on Information Theory*, vol. 25, pp. 219–221, March 1979.

[33] T. Philosof and R. Zamir, "On the loss of single-letter characterization: The dirty multiple access channel," *IEEE Transactions on Information Theory*, vol. 55, pp. 2442–2454, June 2009.

[34] T. Philosof, R. Zamir, U. Erez, and A. J. Khisti, "Lattice strategies for the dirty multiple access channel," *IEEE Transactions on Information Theory*, vol. 57, pp. 5006–5035, August 2011.

[35] I.-H. Wang, "Approximate capacity of the dirty multiple-access channel with partial state information at the encoders," *IEEE Transactions on Information Theory*, vol. 58, pp. 2781–2787, May 2012.

[36] D. Krithivasan and S. S. Pradhan, "Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function," *IEEE Transactions on Information Theory*, vol. 55, pp. 5268–5651, December 2009.

[37] D. Krithivasan and S. Pradhan, "Distribued source coding using Abelian group codes," *IEEE Transactions on Information Theory*, vol. 57, pp. 1495–1519, March 2011.

[38] A. B. Wagner, "On distributed compression of linear functions," *IEEE Transactions on Information Theory*, vol. 57, pp. 79–94, January 2011.

[39] D. N. C. Tse and M. A. Maddah-Ali, "Interference neutralization in distributed lossy source coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2010)*, (Austin, TX), June 2010.

[40] Y. Yang and Z. Xiong, "Distributed compression of linear functions: Partial sum-rate tightness and gap to optimal sum-rate," *IEEE Transactions on Information Theory*, vol. 60, pp. 2835–2855, May 2014.

[41] U. Niesen and P. Whiting, "The degrees-of-freedom of compute-and-forward," *IEEE Transactions on Information Theory*, vol. 58, pp. 5214–5232, August 2012.

[42] Y. Song and N. Devroye, "Lattice codes for the Gaussian relay channel: Decode-and-forward and compress-and-forward," *IEEE Transactions on Information Theory*, vol. 59, pp. 4927–4948, September 2013.

[43] U. Niesen, B. Nazer, and P. Whiting, "Computation alignment: Capacity approximation without noise accumulation," *IEEE Transactions on Information Theory*, vol. 59, pp. 3811–3832, 2013 2013.

[44] Z. Ren, J. Goseling, J. H. Weber, and M. Gastpar, "Maximum throughput gain of compute-and-forward for multiple unicast," *IEEE Communication Letters*, vol. 18, pp. 1111–1113, July 2014.

[45] G. Bresler, A. Parekh, and D. Tse, "The approximate capacity of the many-to-one and one-to-many Gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 56, pp. 4566–4592, September 2010.

[46] A. S. Motahari, S. Oveis-Gharan, M.-A. Maddah-Ali, and A. K. Khandani, "Real interference alignment: Exploiting the potential of single antenna systems," *IEEE Transactions on Information Theory*, vol. 60, pp. 4799–4810, August 2014.

[47] U. Niesen and M. A. Maddah-Ali, "Interference alignment: From degrees-of-freedom to constant-gap capacity approximations," *IEEE Transactions on Information Theory*, vol. 59, pp. 4855–4888, August 2013.

[48] I. Shomorony and S. Avestimehr, "Degrees of freedom of two-hop wireless networks: Everyone gets the entire cake," *IEEE Transactions on Information Theory*, vol. 60, pp. 2417–2431, May 2014.

[49] A. Padakandla, A. G. Sahebi, and S. S. Pradhan, "An achievable rate region for the 3-user interference channel based on coset codes," *IEEE Transactions on Information Theory*, vol. 62, pp. 1250–1279, March 2016.

[50] X. He and A. Yener, "Providing secrecy with structured codes: Tools and applications to two-user Gaussian channels," *IEEE Transactions on Information Theory*, vol. 60, pp. 2121–2138, April 2014.

[51] G. Bagherikaram, A. S. Motahari, and A. K. Khandani, "On the secure DoF of the single-antenna MAC," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2010)*, (Austin, TX), June 2010.

[52] S. Vatedka, N. Kashyap, and A. Thangaraj, "Secure compute-and-forward in a bidirectional relay," *IEEE Transactions on Information Theory*, vol. 61, pp. 2531–2556, May 2015.

[53] J. Xie and S. Ulukus, "Secure degrees of freedom of one-hop wireless networks," *IEEE Transactions on Information Theory*, vol. 60, pp. 3359–3378, June 2014.

[54] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Transactions on Information Theory*, vol. 42, pp. 1152–1159, July 1996.

[55] H.-A. Loeliger, "Averaging bounds for lattices and linear codes," *IEEE Transactions on Information Theory*, vol. 43, pp. 1767–1773, November 1997.

[56] G. Forney, M. Trott, and S.-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol. 46, pp. 820–850, May 2000.

[57] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, pp. 1250–1276, June 2002.

[58] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Transactions on Information Theory*, vol. 51, pp. 3401–3416, October 2005.

[59] O. Ordentlich and U. Erez, "A simple proof for the existence of "good" pairs of nested lattices," *IEEE Transactions on Information Theory*, vol. 62, to appear 2016.

[60] R. Zamir, *Lattice Coding for Signals and Networks*. Cambridge University Press, 2014.

[61] B. Nazer and R. Zamir, *Lattice Coding for Signals and Networks*, ch. Gaussian Networks. Cambridge University Press, 2014.

[62] J. Zhu and M. Gastpar, "Multiple access via compute-and-forward," *IEEE Transactions on Information Theory*, Submitted July 2014. Available online: http://arxiv.org/abs/1407.8463.

[63] S.-N. Hong and G. Caire, "On interference networks over finite fields," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4902–4921, 2014.

[64] S.-N. Hong and G. Caire, "Virtual full-duplex relaying with half-duplex relays," *IEEE Transactions on Information Theory*, vol. 61, pp. 4700–4720, September 2015.

[65] S.-N. Hong and G. Caire, "Structured lattice codes for $2 \times 2 \times 2$ MIMO interference channel," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2013)*, (Istanbul, Turkey), July 2013.

[66] Y. Tan, X. Yuan, S.-C. Liew, and A. Kavcic, "Asymmetric compute-and-forward: Going beyond one hop," in *Proceedings of 52nd Annual Allerton Conference on Communications, Control and Computing*, (Monticello, IL), October 2014.

[67] B. Nazer, V. Cadambe, V. Ntranos, and G. Caire, "Expanding the compute-and-forward framework: Unequal powers, signal levels, and multiple linear expanding the compute-and-forward framework: Unequal powers, signal levels, and multiple linear combinations," *arXiv e-prints*, 2015. Available online: http://arxiv.org/pdf/1504.01690v2.pdf.

[68] N. E. Tunali, K. R. Narayanan, J. J. Boutros, and Y.-C. Huang, "Lattices over eisenstein integers for compute-and-forward," in *Proceedings of the 50th Annual Allerton Conference on Communications, Control, and Computing*, (Monticello, IL), October 2012.

[69] G. D. Forney and D. J. Costello, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, pp. 1150–1177, June 2007.

[70] A. Padakandla and S. Pradhan, "Achievable rate region based on coset codes for multiple access channels with states," *IEEE Transactions on Information Theory*, Submitted January 2013. Available online: http://arxiv.org/abs/1301.5655.

[71] B. Nazer, A. Sanderovich, M. Gastpar, and S. Shamai (Shitz), "Structured superposition for backhaul constrained cellular uplink," in *Proceedings of the International Symposium on Information Theory (ISIT 2009)*, (Seoul, South Korea), June 2009.

[72] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. New York, NY: Springer-Verlag, 1997.

[73] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity of the Gaussian two-way relay channel to within $1/2$ bit," *IEEE Transactions on Information Theory*, vol. 56, pp. 5488–5494, November 2010.

[74] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge: Cambridge University Press, 2005.

[75] M. Nokleby, B. Nazer, B. Aazhang, and N. Devroye, "Relays that cooperate to compute," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS 2012)*, (Paris, France), September 2012.

[76] M. Varanasi and T. Guess, "Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the Gaussian multiple-access channel," in *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), November 1997.

[77] M. R. Bremner, *Lattice Basis Reduction*. CRC Press, 2012.

[78] T. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley-Interscience, 2nd ed., 2006.

[79] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[80] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Transactions on Information Theory*, vol. 42, pp. 364–375, March 1996.

[81] L. Wang, E. Sasoglu, and Y.-H. Kim, "Sliding-window superposition coding for interference networks," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2014)*, (Honolulu, HI), July 2014.

[82] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, pp. 49–60, January 1981.

[83] H. Sato, "The capacity of the Gaussian interference channel under strong interference," *IEEE Transactions on Information Theory*, vol. 27, pp. 786–788, November 1981.

[84] M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Transactions on Information Theory*, vol. 54, pp. 3457–3470, August 2008.

[85] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the K-user interference channel," *IEEE Transactions on Information Theory*, vol. 54, pp. 3425–3441, August 2008.

[86] P. Chebyshev, "Mémoire sur les nombres premiers," *Journal de mathématiques pures et appliquées*, vol. 1, pp. 366–390, 1852.

[87] J. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1998.

[88] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[89] I. E. Bakoury, W. He, and B. Nazer, "Integer-forcing interference alignment: Iterative optimization via aligned lattice reduction," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2015)*, (Hong Kong, China), June 2015.

[90] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, 2nd ed., 2013.

[91] J. W. S. Cassels, *An Introduction to Diophantine Approximations*. Cambridge University Press, 1957.

[92] J. J. Sylvester, "On the relation between the minor determinants of linearly equivalent quadratic functions," *Philosophical Magazine*, vol. 1, no. 4, pp. 295–305, 1851.